2004

Validation of PET/CT dataset for radiation treatment planning

Rajesh Manoharan
Louisiana State University and Agricultural and Mechanical College, rmanoh1@lsu.edu

VALIDATION OF PET/CT DATASET FOR RADIATION TREATMENT PLANNING

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Department of Physics and Astronomy

by
Rajesh Manoharan
B.E, Anna University, 1999
M.S, University of Toledo, 2002
December 2004

To my father, mother and sister

ACKNOWLEDGEMENTS

I would like to thank Dr. Bujenovic for the opportunity that he gave me to work in Lake PET imaging center. It was a good learning opportunity for me in imaging under his able guidance. He was my major advisor and gave tremendous input for my thesis. I also thank
Dr. Oscar for all time he had spent with us helping to learn Radiation therapy physics. He was helpful in building the experimental setup.

I also thank Dr. Matthews for teaching imaging physics and his able guidance for the thesis. He was helping in honing my thoughts and fine tuning the work. I thank Dr.Sajo and his student Aimee Verrette for making ^{22}Na seeds. Without the seeds they offered this experiment would never be successful. I also thank Dr. Butler for his help in Mathematica programming. His class on 3D image analysis was really helpful in doing image analysis for this thesis. I also thank Dr. Frank for being part of my thesis committee and reviewing my work.

I thank Yuri Ishihara for her support as a friend and co-author during my initial work. Thanks to Ron, Ryan and Darren for their help in familiarizing me with the operation of the PET/CT scanner. Last and most thanks to all my medical physics friends who have been a motivation and support during my course of study.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS...iii

LIST OF TABLES vi

LIST OF FIGURES ...viii

ABSTRACT...x

CHAPTER 1. INTRODUCTION 1
1.1 Introduction 1
1.2 Background 3
1.2.1 Radiation Treatment Planning 3
1.2.2 Computed Tomography Imaging 6
1.2.3 Positron Emission Tomography Imaging 9
1.2.4 Dual Modality PET/CT Imaging 13

CHAPTER 2. DEFINITION OF THE PROBLEM 15
2.1 Quality Assurance in PET/CT 15
2.2 Literature Review 16
2.3 Definition of the Problem 18

CHAPTER 3. MATERIALS 21
3.1 PET/CT Scanner 21
3.2 Phantom 22
3.3 Software 24
3.3.1 Acquisition and Fusion Software 24
3.3.2 Computation Software 25
3.3.3 RTP Software 25

CHAPTER 4. METHODS 26
4.1 Preparation of Radioactive Sources 26
4.1.1 Making of ^{22}Na Seeds 26
4.1.2 Preparation of Point Source 27
4.2 PET/CT Data Acquisition 27
4.2.1 Phantom Setup 27
4.2.2 Input and Reconstruction Parameters 28
4.3 Data Analysis using Syngo Software 30
4.3.1 CT Number Analysis....30
4.3.2 Geometric Scaling Analysis 30
4.3.3 Registration Error Analysis 33
4.4 Data Analysis in RTP System 35
4.5 Data Analysis using Mathematica Computation Software 36
4.5.1 Generation of One-Dimensional Image Plot from Two-Dimensional Images 36
4.5.2 Nonlinear Gaussian Curve Fitting Procedure 39

4.5.3 Determination of Centroid in PET and CT Images ……41
4.5.4 Registration Check for PET/CT Fused Image ……44
4.5.5 Geometric Scaling Accuracy Check ……45
4.6 NEMA Resolution Analysis ……45

CHAPTER 5. RESULTS AND DISCUSSION ……49
5.1 CT Number Analysis……49
5.2 Geometric Scaling Analysis ……....50
5.3 Registration Error Analysis ……57
5.4 Resolution Analysis ……62

CHAPTER 6. CONCLUSION ……64

REFERENCES…..……69

APPENDIX A. CTPIXEL.NB……...72

APPENDIX B. FUS.NB……80

APPENDIX C. PETCTFUSION.NB……88

APPENDIX D. TRANSRES.NB……107

VITA ……..122

LIST OF TABLES

1) Manufacturer's resolution specification for the REVEAL HD PET/CT Scanner ……22

2) ^{22}Na seed activities ……26

3) PET and CT acquisition parameters ……29

4) Reconstruction parameters ……30

5) CT number value for electron density insert ……31

6) Formulae for computing spatial resolution report values ……48

7) Measured CT number values for 5 mm slice thickness ……49

8) Measured CT number values for 1 mm slice thickness ……49

9) Measured CT number value in RTP system for 5 mm slice thickness ……50

10) Measured CT number value in RTP system for 1 mm slice thickness ……51

11) Measured cube dimensions for 5 mm CT slice thickness ……52

12) Measured cube dimensions for 1 mm CT slice thickness ……52

13) Measured distance in PET image between ^{22}Na seeds 10 cm apart ……52

14) Measured cube dimensions for 5 mm CT slice thickness in RTP system ……53

15) Measured cube dimensions for 1 mm CT slice thickness in RTP system ……53

16) Number of pixels in 10 cm distance for CT slice thickness of 5 mm ……...54

17) Number of pixels in 10 cm distance for CT slice thickness of 1 mm ……...54

18) Measured pixel size for CT matrix of 5 mm slice thickness ……54

19) Measured pixel size for CT matrix of 1 mm slice thickness ……55

20) Number of pixels for 10 cm distance in PET image ……56

21) Measured pixel size for PET matrix ……56

22) Number of pixels in PET only fused image for 10 cm ……56

23) Number of pixels in CT only fused image for 10 cm …….. 57

24) Measured pixel size for PET and CT only fused image matrixes ……57

25) Computed centroid pixels in PET only fused image ……58

26) Computed centroid pixels in CT only fused image ……..58

27) Computed registration error in pixels in X and Y direction …….60

28) Computed registration error in PET/CT fused image in pixels and mm ……..60

29) Measured FWHM in pixels at different point based on NEMA 2-2001 ……..60

30) Measured registration error of PET/CT dataset in RTP system ……...61

31) Measured resolution based on NEMA 2001 …….62

32) Comparison of specified resolution and measured resolution ……..63

LIST OF FIGURES

1) ICRU recommended treatment volume …….5

2) Helical scanning ……..7

3) Helical interpolation ……8

4) Annihilation event ……...9

5) Annihilation coincidence circuitry and PET detector geometry ……...10

6) Coincidence events ……...11

7) Attenuation correction factor (ACF) …….12

8) PET/CT scanner ……....13

9) REVEAL HD scanner ……..21

10) TGM^2 ISIS QA-1 Geometric QA Phantom ……..23

11) Set up of TGM^2 QA Phantom on flat patient table ……...28

12) Syngo CT number measurement ……...31

13) Syngo dimension analysis …….32

14) Windowing and leveling by adjusting contrast window ……...33

15) PET and CT windows ……...34

16) Flowchart of finding registration error using Syngo 3D fusion tool ……35

17) Binary CT phantom image ……...37

18) Cropped PET phantom image ……..38

19) Summation of 2D gray scale image to 1D to find column and row sums ……...38

20) One dimensional PET image plot of the ^{22}Na seeds ……39

21) Mathematica function for nonlinear Gaussian curve fit ……..40

22) Result of nonlinear Gaussian curve fit …… 41

23) Flowchart of finding centroid and FWHM in PET image ……...42

24) Flowchart to find centroid and pixel size in CT image ……...43

25) Flowchart of registration error analysis using Mathematica software ……44

26) Styrofoam block inserted with FDG filled capillary tube point sources based on NEMA recommendation ……..46

27) Flowchart of computing resolution based on NEMA recommendation …….47

28) Syngo PET/CT fused image ……59

29) PET/CT fused image in RTP system using Syntegra …….61

30) Fused image of TGM^2 phantom with PET object insert in RTP system ……62

ABSTRACT

PET/CT scans are frequently used for radiation treatment planning (RTP). Our work demonstrates a practical approach for validating the PET/CT dataset for RTP. We tested this QA process on a Reveal HD PET/CT scanner. The phantom used is a TGM^2 ISIS QA phantom, a 14 cm acrylic cube with a central bore for object inserts. It has four different built-in inserts for electron density verification. ^{22}Na seeds are inserted into the pinholes at the side of the cube. PET/CT images of the phantom with ^{22}Na seeds are acquired and fused in the scanner Syngo fusion software. Registration of the PET/CT dataset is visualized by raising the lower threshold of the PET images to reduce the ^{22}Na point sources to a few pixels and comparing it with the CT images of ^{22}Na seeds. Geometric scaling accuracy of the pixels is verified by measuring the dimension of the cube in x, y and z axes. The HU values of four electron density verification inserts are measured and compared with manufacturer specified HU values. These QA tests are repeated in the RTP software after importing the PET/CT dataset. A quantitative analysis of registration error and geometric scaling accuracy of pixels are verified independently using MATHEMATICA. The resolution of the PET scanner was determined by measuring the FWHM of capillary tube sources inserted in a Styrofoam block based on the NEMA-2 protocol.

Minor misalignment of the fused images was detected in the scanner (~1 mm) while the imported dataset in the RTP system showed a major misalignment (~6 mm) when fused by auto fusion software. The maximum geometric scaling errors of object sizes were observed in the z direction (5.2% decrease) in the scanner and the scaling errors were less in the RTP software (2.9% decrease). The greatest HU errors in the CT image compared with expected HU values were observed in the bone density insert (28% increase) in the scanner and all HU values for

different inserts were shifted up by a constant value in the RTP system. The resolution of the PET scanner was comparable to the manufacturer's specification.

CHAPTER 1
INTRODUCTION

1.1 Introduction

Diagnosing, staging, and re-staging of cancer, as well as the monitoring and planning of cancer treatment, has traditionally relied on anatomic imaging like computed tomography (CT) and magnetic resonance imaging (MRI). Spatially accurate medical imaging is an essential tool in three dimensional conformal radiation therapy (3DCRT) and intensity-modulated radiation therapy (IMRT) treatment planning. CT imaging is the standard imaging modality for image-based radiation treatment planning (RTP). CT images provide anatomical information on the size and location of tumors in the body. They also provide electron density information for heterogeneity-based patient dose calculation. The major limitation of the CT imaging process is soft tissue contrast, which is overcome by using contrast agents or using another anatomical imaging modality like MRI.

One of the disadvantages of anatomical imaging techniques like CT and MRI is its inability to characterize the tumor. Tumors need to be characterized whether they are benign or malignant and if malignant it would be helpful to know whether the proliferation is slow or fast. Necrotic, scar, and inflammatory tissue often cannot be differentiated from malignancy based on anatomic imaging alone. Anatomical imaging has high sensitivity for detection of structural changes, but a low specificity for further characterization of these abnormalities. Single photon emission computed tomography and positron emission tomography (PET) are imaging techniques that provide information on physiology rather than anatomy. These modalities have been used for evaluation of tumor metabolism, differentiation between tumor reoccurrence and radiation necrosis, detection of hypoxic areas of the tumor, and other functional imaging.

Radiation treatment planning requires an accurate location of the tumor and the normal tissue and also knowledge of the size of the tumor for contouring the treatment volume. Although PET provides necessary functional information for RTP, it has a few limitations. The spatial resolution of PET is too poor to give accurate quantitative information. The greatest limitation in using PET for RTP is its lack of anatomical information. This limitation of PET is overcome by evaluating PET and CT images together. Fused PET and CT images give better diagnostic evaluation than PET or CT images used alone [1, 2]. But fusion of PET and CT images are meaningful only when they are correctly spatially registered. Hence a proper spatial registration is required for accurate delineation of tumor volume.

The necessity of accurate spatial registration of fused images requires different fusion techniques for different image datasets. Software fusion and hardware fusion are the two different approaches considered by the scientific community [3, 4]. Software fusion approaches use different transformation algorithms to fuse different modality images acquired at different times. The transformation algorithms are classified as rigid and nonrigid transformation algorithms. They are based on whether they fuse images of rigid-body (e.g., head) or non rigid (e.g., abdomen) objects [5, 6]. Although software fusion gives better diagnostic information than using separate images, physicians may not rely on the information if the fused images were acquired at different times. Also the chances of a change in patient position are high for image acquisition done at different times.

The hardware approach of image fusion is headed towards designing a single imaging system to acquire simultaneously the different image modalities required [3]. Hardware fusion is partially achieved by construction of a hybrid PET/CT scanner [7, 8] which acquires different modalities sequentially. These hybrid scanners are two separate scanners enabled to operate in

sequence one after another to acquire the different image modality datasets in a single imaging session. Although hybrid scanners do not give a true hardware fusion and have not proven to be a better fusion technique scientifically [9], they have gained popularity for image acquisition in a single session. Due to reduced scan time and patient motion, PET/CT is considered reliable among the oncology community.

These hybrid PET/CT scanners, due to reduced scan time and reliable registration of PET and CT datasets, are becoming common in RTP. A PET image fused with a CT image can be used in treatment planning to eliminate geographic misses of the tumor and escalation of dose to the hypermetabolic aspects of a tumor. Fused images improve the accuracy in staging of lymph nodes [10, 11]. Although the use of PET/CT in RTP is growing at a fast pace, little research has been done in the direction of validating the PET/CT datasets for RTP. This thesis discusses the current research in validation and suggests a methodology to validate PET/CT datasets for RTP using a phantom and image analysis software.

1.2 Background

In the remaining sections in this Chapter the background information related to this thesis is discussed. The basic procedures in RTP and principles behind the CT imaging, PET imaging and PET/CT imaging are explained.

1.2.1 Radiation Treatment Planning

Planning for radiation therapy begins with the process of defining and localizing the volume of tissue to be irradiated. Traditionally localizing the volume of tissue to be irradiated and normal tissue to be protected was done on planar X-ray film images. Tumor volume and normal structures were contoured on radiographic films. Once the treatment volume is defined radiation beams are designed to surround the tumor volume while sparing the normal structures. Then

treatment simulation is done to validate the delivery of the prescribed radiation dose to the target volume. Treatment simulation mostly checks the orientation of the beams and their size, the placement of field shaping blocks, and the placement of marks on the patient to allow for reliable reproduction of treatment geometry every day. This simulation was traditionally done using a conventional simulator which is similar to a treatment machine but having a diagnostic X-ray tube instead of the megavoltage radiation source. It was equipped with fluoroscopic display enabling the operator to view and modify the beam placement in real time. A simulator radiographic film is shot to record the beam placement. Finally a portal film is shot in the treatment machine so patient position can be compared to the simulator radiographic film for verification of treatment setup before treatment.

In modern RTP the planar radiographic films are replaced by CT images. CT imaging improves the accuracy of localization of diseased tissue and normal structures. It also replaces the conventional simulator to produce the CT dataset now used for treatment simulation. Most treatment planning software used today are equipped with the beams eye view (BEV) option to design beam shape and with digitally reconstructed radiograph (DRR) to replace simulator films. Although the tools for treatment planning have changed, the basic treatment volume delineation process mostly remains the same. ICRU-recommended nomenclature for treatment volume delineation is shown in Figure 1. It identifies five levels in delineation of target volume, the gross tumor volume (GTV), clinical target volume (CTV), planning target volume (PTV), treated volume and irradiated volume. GTV is described as the visible location and extent of the malignant growth. This volume together with sub-clinical microscopic malignant disease makes the CTV. PTV includes the CTV plus margins for patient motion, organ motion, organ shape and size variation, and uncertainties in beam placement. Treated volume is the volume enclosed by

the isodose surface representing minimum target dose. The minimum target dose is prescribed for PTV. Due to the limitations of treatment techniques additional margins are provided to PTV to adequately cover PTV with minimum target dose. The treated volumes are generally larger than PTV. Irradiated volume is the tissue receiving 50% of the specified target dose. The geometric field size of the beam coincides with the 50% isodose line at reference depth. The irradiated volume is larger than the treated volume. These are the common concepts and terms used in defining target volume as recommended by ICRU.

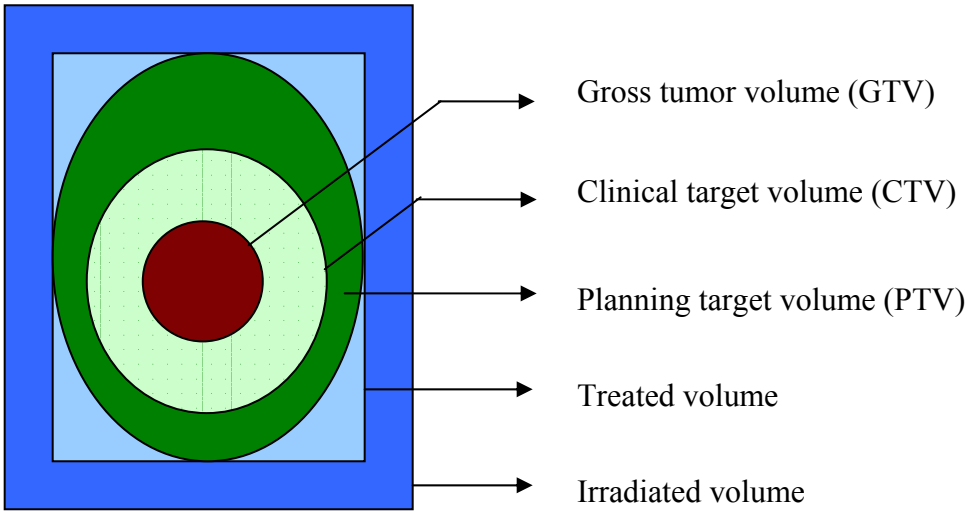


Fig. 1: ICRU recommended treatment volume

The goal of RTP is to deliver a lethal dose to the tumor. To achieve more accuracy in delivering this dose, new technologies like 3DCRT and IMRT have been introduced. These techniques create a steep dose gradient around the tumor volume allowing the escalation of the dose for tumor without increasing dose to adjacent healthy organs. These new techniques strictly depend on the imaging modality's ability to characterize and localize the tumor in the best possible way. PET imaging which provides metabolic information improves the knowledge of disease and treatment design. PET images fused with CT images can be incorporated in RTP.

More oncologists are using information obtained from PET imaging to design appropriate GTV for the tumor region. Correct co-registration of PET and CT images is essential for optimum treatment design. To ensure proper delivery of dose by these modern radiotherapy techniques, validation of PET and CT datasets for RTP is necessary.

1.2.2 Computed Tomography Imaging

CT images describe the electronic density distribution of cross sections of the patient anatomy. CT systems provide gray scale display of linear attenuation coefficients that closely relate to the density of the tissue. CT imaging evolved from conventional planar radiographs. In planar X-ray film imaging the three dimensional anatomy of the patient is reduced to a two dimensional attenuation projection image and the depth information of the structures are lost. In CT imaging several attenuation projection images for a volume of tissue are acquired at different angles. These sets of projection images are reconstructed by filtered backprojection algorithm to generate two dimensional attenuation cross-section of anatomy of the patient. The attenuation measurement for a CT detector element is given by Equation 1 and Equation 2. Equation 1 represents attenuation measurement for homogenous object and Equation 2 represent attenuation measurement for inhomogeneous (heterogeneous) objects.

$$P(\bar{x}) = \ln[\frac{I_o}{I(\bar{x})}] = \mu x \qquad (1)$$

$$P(\bar{x}) = \ln[\frac{I_o}{I(\bar{x})}] = \int_L \mu(\bar{x}) d\bar{x} \qquad (2)$$

where $P(\bar{x})$ is the measured projection data for attenuation along the $\bar{x}$ direction. I_o is the intensity of the x-ray beam measured without the patient in the way for that detector element. This is also known as a blank scan as described in Chapter 2. $I(\bar{x})$ is the measured intensity after

attenuation by the patient. $\mu(\bar{x})$ is the measured attenuation coefficient as a function of location in the patient.

A CT scanner positions a rotating x-ray tube and detector on opposite sides of the patient to acquire projection images. Early CT scanners used pencil beams of x-rays and a combination of translation and rotation motion to acquire projection images [12]. Modern CT scanners have a stationary or rotating detector array with a rotating fan beam x-ray tube. There are also two types of scanning: axial and helical CT scanning. In axial scanning the patient is moved step by step acquiring sets of projection images for each slice. In helical scanning (Fig 2) the patient table moves continuously while the x-ray tube acquires a series of projection images [13]. The projection images are acquired for a helical path around the patient. In helical scanning to reconstruct a cross-sectional planar image, the helical data is interpolated to give axial plane projection data before reconstruction (Fig 3). By removing the time to index the table between slices the total scan time of the patient is reduced. Also reconstruction can be done for any slice thickness after acquiring the data. This helical scanning is available in most of the current CT scanners.

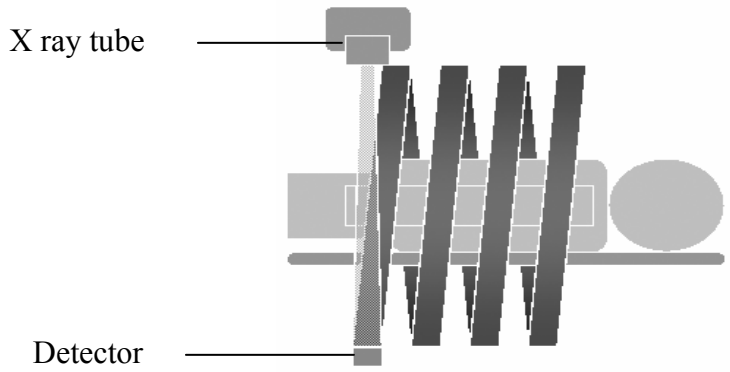


Fig 2: Helical scanning
(Continuous scanning while table moves)

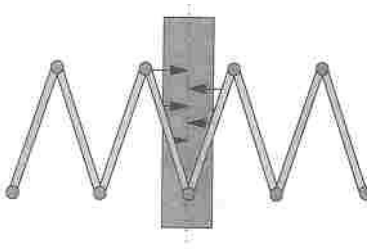


Fig 3: Helical interpolation

The reconstructed CT image is a two dimensional matrix of numbers, with each pixel corresponding to a spatial location in the image and in the patient. Usually the matrix is 512 pixels wide and 512 pixels tall covering a 50 cm x 50 cm field of view. The numeric value in each pixel represents the attenuation coefficient as a gray level in the CT image. These numbers are called Hounsfield units or CT numbers. The reconstruction process generates a matrix of Hounsfield units which give the linear attenuation values normalized to the attenuation of water. This normalization is given by Equation 3.

$$\text{CT Number (HU)} = 1000\frac{(\mu_{pixel} - \mu_{water})}{\mu_{water}} \qquad (3)$$

CT number gives an indication of the type of tissue. Water has a CT number of zero. Negative CT numbers are typical for air spaces, lung tissues and fatty tissue. Values of μ_{pixel} greater than μ_{water} correspond to other soft tissues and bone.

Radiologists occasionally make critical diagnostic decisions based on CT number of particular regions of interest. Also attenuation values given by CT numbers are used to calculate the dose delivered to the tumor in RTP. CT number is an important parameter in CT images which must be frequently checked for accuracy.

1.2.3 Positron Emission Tomography Imaging

Positron emission tomography (PET) imaging generates images that depict the distribution of positron emitting radionuclide in the patient body. PET imaging often uses the F-18 fluorodeoxyglucose (FDG) radioactive tracer to track increased glucose metabolic activity of tumor cells and to provide images of the whole body distribution of FDG. When the positron is emitted by the radioactive tracer it annihilates with an electron to generate two 511 kev photons emitted in nearly opposite directions (Fig 4). These photons interact with the ring of detector elements surrounding the patient (Fig 5). If both the emitted photons are detected then the point of annihilation lies on the line joining the points of detection. This line joining the points of detection is known as the line of response (LOR). The circuit used by the scanner to record the detector interactions occurring at the same time is called coincidence circuitry. This whole process is called annihilation coincidence detection. Thus a PET scanner uses annihilation coincidence detection instead of mechanical collimation like gamma cameras to acquire projections of activity distribution in the patient. Projections acquired at different angles are reconstructed using iterative algorithms to generate cross-sectional images of activity distribution.

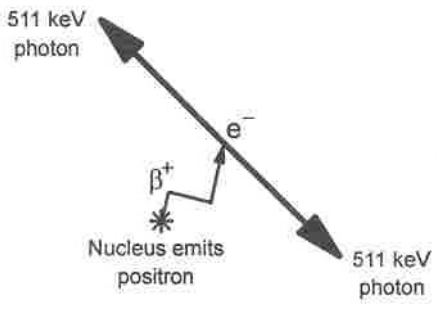


Fig 4: Annihilation event

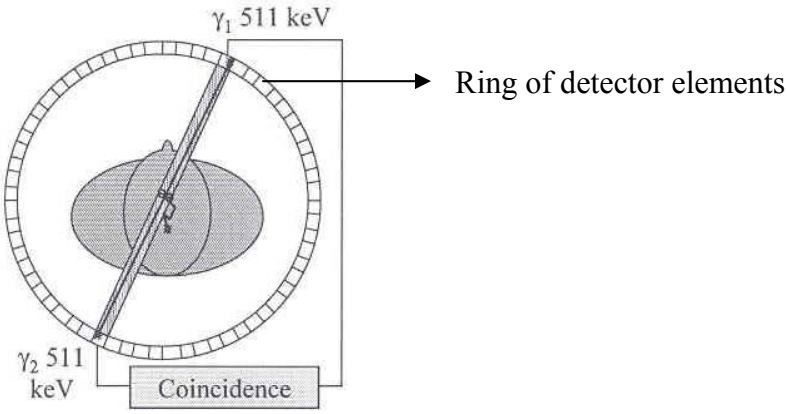


Fig 5: Annihilation coincidence circuitry and PET detector geometry

The annihilation coincidence detection process allows many false events to be acquired. Corrections are necessary for these false events before the projections are reconstructed. The total events acquired are classified as trues, random and scatter (Fig 6). A true coincidence is simultaneous interactions occurring in the detectors resulting from emissions occurring in the same nuclear transformation. Random coincidences occur when emissions from different nuclear transformations interact in coincidence with the surrounding detectors. Scatter coincidence occurs when one or both photons from annihilation is scattered in the patient body and interact with the detector to give a false LOR. The acquired annihilation events need to be corrected for random and scatter events. Random coincidence events along any LOR may be directly measured using the delayed coincidence method [14]. The delayed coincidence method uses two coincidence circuits. The first circuit measures both true and random coincidence events. The second circuit has a delay of several hundred microseconds inserted into the coincidence window, so all true coincidences are thrown out of coincidence. The counts measured in the second circuit are subtracted from the first to give true counts. Scatter correction is done for the projection data by model-based scatter estimation [14]. The scatter correction

factor is estimated by mathematical models and applied to the projection data before reconstruction.

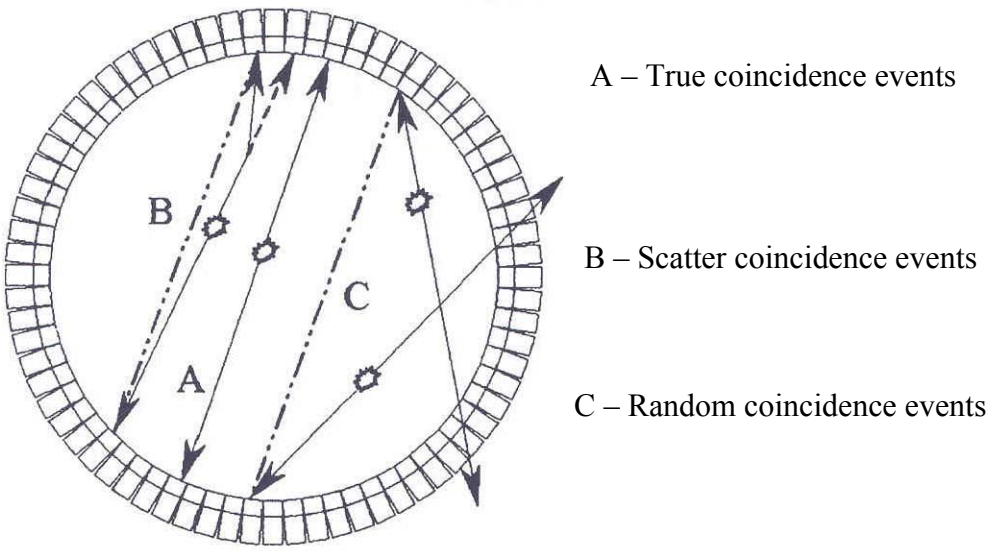


Fig 6: Coincidence events

Annihilation photons emitted by nuclear transformation are attenuated by the patient body. Hence correction for attenuation is also necessary to get an accurate activity distribution. The probability of both photons escaping is the product of the probabilities of each escaping. The probability of photons interacting within the coincidence window is proportional to the probability of photons escaping [15-17]. The total escape probability is given by Equation 6 and Fig 7 where P_1 and P_2 in Equation 5 and Equation 6 are the escape probabilities of each photon. The attenuation correction factor (ACF) is nothing but the inverse of the total escape probability that is measured from the transmission ratio $I_o/I(x)$ along the LOR. ACFs are measured by using a rotating positron source or other photon source. These photon emitting sources are called transmission sources; they rotate around the patient inside the detector ring to measure the transmission of photons through the patient. ACFs correct for the attenuation of annihilation photons in the patient as shown in Equation 7. Attenuation measured by photon sources of

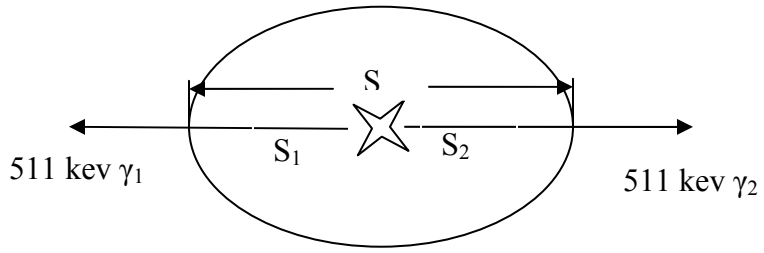


Fig 7: Attenuation correction factor (ACF)

$$P_1 = e^{-\int_{S_1} \mu(\bar{s})\, d\bar{s}} \quad (4)$$

$$P_2 = e^{-\int_{S_2} \mu(\bar{s})\, d\bar{s}} \quad (5)$$

$$P_{esc} = P_1.P_2 = e^{-\int_{S_1+S_2} \mu(\bar{s})\, d\bar{s}} = \frac{I(\bar{s})}{I_0} \quad (6)$$

$$ACF = e^{\int_{S_1+S_2} \mu(\bar{s})\, d\bar{s}} = \frac{I_o}{I(\bar{s})} \quad (7)$$

I_o is the number of non attenuated photons detected for a detector pair along a LOR

$I(\bar{s})$ is the number of photons detected for a detector pair after attenuation along the LOR

energy other than 511 kev are scaled to 511 kev ACFs. Usually in the stand-alone PET scanner these transmission scans take most of the time in PET data acquisition [15, 16].

1.2.4 Dual Modality PET/CT Imaging

Image fusion was initially achieved by software fusion of anatomical and functional images. Software fusion was generally successful with brain and rigid body volumes. It encountered significant difficulties when fusing images of the rest of the body. Alignment algorithms fail to converge the two image sets due to problems of patient movement or discrepancies in patient positioning between two scans. Also involuntary movements of internal organs arise when patient are imaged on different scanners and at different times. Dual modality PET/CT imaging is a combination of imaging technologies helping to acquire accurately aligned anatomical and functional images in the same scanning session (Fig 8). Also an additional advantage of the combined PET/CT scanner is the use of CT images for attenuation correction. CT images can be scaled in energy and used to correct the PET data for attenuation effects [16, 17]

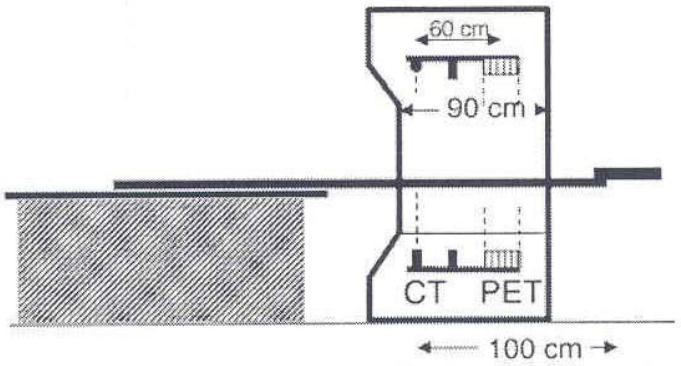


Fig 8: PET/CT scanner

Dual-modality PET/CT was first built at the University of Pittsburgh in collaboration with CTI (Knoxville, TN) and Siemens Medical Solutions (Hoffman Estates, IL), combining separate PET and CT scanning devices into one device. The PET/CT prototype consisted of a rotating partial ring PET system and a single slice CT scanner mounted on the same rotating support. The CT scanner combined with PET often uses helical scanning CT to enable fast patient throughput, but new scanners with both helical and axial scanning are available now. The CT data is usually

acquired first, followed by PET acquisition. There are typically two separate acquisition processing units for CT and PET, and an integrated display workstation. The acquired CT and PET datasets are send to the reconstruction processing unit for reconstruction. Reconstructed images are fused in the fusion workstation. CT and PET images can also be separately viewed in the workstation.

The protocol for PET/CT imaging starts with patient preparation. 5 – 15 mCi of FDG is injected into the patient 45 – 60 min before the start of image acquisition. After 45 min, the glucose circulates through the body; the patient gets ready for image acquisition by emptying the bladder. The patient is positioned on the table for an initial topogram. The topogram is used to select the scan range for PET/CT image acquisition. The scan range is selected as a number of bed positions. Once the image acquisition region is selected in the topogram, the helical CT scan is done first; it takes around 30 sec to acquire one bed position. After completion of the CT portion, the scanner bed is moved to the PET starting position and the emission scan is started. The emission scan duration per bed position varies with the detector technology used. With conventional bismuth germinate oxyorthosilicate (BGO) system, acquisition times will range from 5 to 8 minutes per bed position. The new lutetium oxyorthosilicate (LSO) technology reduces emission scans to 3 to 5 minutes per bed position [18]. The CT data are used to perform attenuation correction. Image reconstruction is completed a few minutes after the PET image acquisition is completed. Since the CT data is used for attenuation correction, the total scan duration for a PET/CT scanner is shorter than that for stand-alone PET scanner, because the CT acquisition is much faster than a conventional PET transmission acquisition.

CHAPTER 2
DEFINITION OF THE PROBLEM

2.1 Quality Assurance in PET/CT

As PET/CT imaging is gaining grounds in RTP, quality assurance (QA) protocols to check the PET and CT as a combined device are needed. QA for PET/CT is not well defined and there is research in progress to define a standard methodology for checking their performance. Some research work has analyzed the artifacts of helical CT and its impact on the attenuation correction of PET images. These studies analyze the problems faced in PET/CT imaging due to breathing and to artifacts from metal and oral contrast agents [19-22]. Most of the QA currently done in PET/CT scanner facilities are based on stand-alone PET and CT QA protocols.

PET scanner quality control includes system corrections such as normalization, calibration and blank scans. Normalization correction compensates for variation in efficiency in each line of response (LOR) in the sinogram. Calibration correction is used to convert the reconstructed image pixel values into activity concentrations. Both normalization and calibration correction are used to compensate for sensitivity variation in the scanner [23, 24]. The blank scan typically is acquired daily using a transmission rod source and it is used with patient transmission data to obtain ACFs. In PET/CT scanners only the normalization and calibration scans need to be done; a blank scan is not needed as CT scans provide ACFs. Blank scans are done during regular CT daily QA instead of the transmission source blank scan used in stand-alone PET scanners. Blank scans give the number of unattenuated photons (I_o) reaching a detector from the x-ray tube. When used with the number of photons detected (I) by a detector during CT imaging it gives the ACF for CT energy photons.

Daily and monthly QA for PET imaging in PET/CT scanners is done by scanning a uniform ^{68}Ge cylindrical phantom for a normalization scan. Both normalization and calibration corrections are checked. The reconstructed PET images are checked for variation in uniformity. This is done by checking the chi-square value of the acquired data. The CT QA is done by checking CT number for an electron density phantom. The CT phantom is a cylindrical hollow acrylic phantom filled with water. Once CT images of the phantom are taken the CT number values of acrylic, water and air are checked and recorded. Thus current QA methods for PET/CT are more oriented to verify the performance as a stand-alone device rather than as a combined device. Hence validation of the PET/CT dataset has to be done in the scanner and also for the imported PET/CT dataset in the RTP software.

2.2 Literature Review

Accuracy of image registration has remained a significant error factor as contours based on mis-registered images can lead to reduced probability of tumor control and increased probability of normal tissue complication. Most of the earlier studies on analyzing registration accuracies were mostly done for stand-alone PET and CT images. Registrations of multimodality images were based on either external markers or mutual information. Image fusion software was developed which used external markers or mutual information. Mutual information based fusion software uses rigid or non-rigid translation algorithms to fuse the datasets [5, 6]. External markers are called fiducial markers; they are mixtures of radio-opaque and radioactive material. These fiducial markers mount on the patient and phantom and they are visible both on CT and PET images, aiding proper alignment.

Early studies were aimed at developing automated registration software without identifying corresponding structures [25, 26]. They used mutual information similarity of PET and MRI

images to register images. The automated fusion method was involved in fusing MRI and PET images of the brain, but it did not give any direct measure to judge the accuracy of registration. Following this, a phantom for QA of multimodality image registration was developed [27]. This phantom has structures visible in image modalities like CT, PET and MRI. A commercial anthropomorphic head phantom used to assess stereotactic localization accuracy was modified to evaluate the registration process. The image registration was done by fusion software and it was verified by fiducial markers mounted on the phantom. The image registration was verified by manual comparison of external markers in the different imaging modalities.

A recent study developed a different methodology of fusion based on mutual information [9]. It used PET transmission images instead of PET emission images for software fusion based on mutual information. It showed improved accuracy in registration when the patients were imaged with radiotherapy treatment masks. Fusions performed without the mask were not better. There was also a very recent study on validation of co-registration of PET and CT images [28]. They verified the registration accuracy of semi-automatic and automated software fusion. These registration accuracy results were based on having fiducial-based registration as the gold standard. They used an International Atomic Energy Association brain phantom and anthropomorphic head phantom for their experiments.

Most of the above studies on registration accuracies were based on fusion of separate PET and CT images. PET/CT hybrid imaging has gained popularity because sequential acquisition and subsequent visualization of PET superimposed on CT may improve diagnostic accuracy. There is not enough scientific evidence to prove this belief [29]. As PET/CT hybrid imaging is growing, now the focus is on checking the registration accuracies for PET/CT hybrid scanners.

Research has started to define QA protocols for PET/CT scanners and their application in RTP [30]. My work is to develop a QA methodology for validating the PET/CT dataset for RTP.

2.3 Definition of the Problem

The main objective of this work is to design a fast and simple QA process to verify and validate the accuracy of the PET/CT dataset for RTP. This QA process will focus on checking the performance of the PET/CT scanner as a combined device rather than as separate devices. The validation QA involves regular PET and CT performance QA and a QA test to check the scanner as a combined device. This QA procedure is tested both in the scanner system and the RTP system. The TGM^2 Geometric QA phantom with ^{22}Na seeds are used to perform a process oriented QA analysis to check the PET/CT dataset in the scanner as well as in the RTP system. This validation QA includes possible regular CT QA with the TGM^2 Phantom, PET resolution analysis, registration of the PET/CT dataset, and geometric scaling analysis of the PET/CT matrix. The resolution of the PET scanner is measured and compared with the manufacturer specification.

The designed validation QA involves the following tasks:

a) Data analysis using Syngo software
    1. CT number analysis
    2. Geometric scaling analysis
    3. Registration error analysis
b) Data analysis using RTP software
    1. CT number analysis
    2. Geometric scaling analysis
    3. Registration error analysis

c) Data analysis using Mathematica software

1. Geometric scaling analysis
2. Registration error analysis
3. NEMA PET resolution analysis

The PET and CT images are simply matrices with number values. These matrices are checked whether they translate physical dimensions accurately. This translation or geometric scaling accuracy is verified by measuring the dimension of the phantom in the scanner as well as in the RTP software. Also the registration accuracy of the PET and CT datasets is verified in the scanner as well as in the RTP system. In the scanner the dimension of the object and registration accuracy are measured using the Syngo acquisition and fusion software. In the RTP system, the Syntegra fusion tool is used to check the geometric scaling and registration accuracy. The performance of the CT scanner is verified by measuring CT numbers of the electron density inserts in the TGM^2 phantom. PET scanner performance is verified by measuring the resolution by NEMA standards [31, 32].

The above listed data analysis QA checks done in the Syngo scanner software and the RTP software are validated by repeating the data analysis QA checks using an independent program written in Mathematica 5. Geometric scaling analysis in Mathematica is done by measuring the pixel size of the image matrices. The geometric scaling accuracy check is done for the PET matrix, CT matrix and the PET/CT fused matrix. The measured pixel size must be compared with the calculated pixel size based on matrix size and field of view. The pixel size is measured by counting the number of pixels between two known points. The distance between the two points when equated with the number of pixels gives the pixel size. Registration accuracy is also verified by checking the alignment of PET and CT images in the fused images. In the fused

image displaying PET only, the centroid of the PET activity is found. Then the centroid is found for the CT only display. The difference in the centroid of CT and PET quantifies the registration misalignment.

The resolution of the system is found by measuring the point spread function of the system. The point spread function is found by measuring the response of the imaging system for a point source. A point source represents a delta function and the spatial resolution of the imaging system is measured by calculating the FWHM of the image of the point source.

CHAPTER 3
MATERIALS

3.1 PET/CT Scanner

The PET/CT scanner used for testing our QA protocol is a REVEAL HD (CPS/Siemens) combined PET/CT scanner (Fig. 9) . The PET is an ECAT (CTI PET Systems) scanner and the CT scanner is a Siemens Somatom Emotion CT [33]. These two scanners are combined together and controlled by high performance CT image control system (ICS) with integrated Windows NT-based Syngo platform software. It has a common graphical user interface for PET and CT. This PET/CT system has 3D PET acquisition and dual slice helical CT scanning. It does not have septa to restrict scatter coincidences. Acquired PET/CT data are stored in a common Syngo database. The imaging session has a sequence of CT topogram, helical CT and PET acquisition. PET positioning and initialization occurs automatically after the helical CT. CT-based data correction is applied to the PET image and parallel acquisition and processing are done.

Fig 9: REVEAL HD scanner

The REVEAL HD PET/CT has a patient port of 70 cm. The PET detector material is BGO of dimension 4.05 x 4.39 x 30 mm^3. It has 64 detectors per block and 4 photomultiplier tubes. It has 32 detector rings with 576 detectors per ring. The axial FOV of the PET is 155 mm and the transaxial FOV is 660 mm. The PET slice thickness is 2.43 mm. About 63 slices of PET images are generated for a single bed acquisition. The coincidence time window is 12 ns and the energy window used is 350 – 650 keV. Clinical PET images are reconstructed using 3D reconstruction methods. The resolution performance characteristics of the PET scanner are given in Table 1.

Table 1: Manufacturer's resolution specification for the REVEAL HD PET/CT Scanner

	Resolution
	Manufacturer specification, mm
At 1 cm radius	
Transverse Resolution	4.5
Axial resolution	4.2
At 10 cm radius	
Transverse Resolution	5.6
Axial resolution	5.7

The CT scanner is a Siemens Somatom Emotion dual slice CT. It is capable of topograms, helical CT and axial CT. The topogram defines the scan range of interest. Then a volume CT scan is acquired by continuous table feed. The overall scanning time ranges from 80 sec to 100 sec for the longest scanning range. The available slice thicknessess are 1, 2, 3, 5, 8, and 10 mm. The transaxial field of view for CT is 500 mm.

3.2 Phantom

A TGM^2 ISIS QA-1 phantom is used for most of the quality checks for this QA project. The ISIS QA-1 Phantom is a 14 cm acrylic cube geometric QA phantom (Fig 10). This phantom was

designed to aid in verifying the geometric laser positioning accuracy of multiple laser systems in a department. Laser position accuracy can be verified in the imaging scanner, simulator and in the treatment accelerator. It has 2 mm wide 10 cm long square grooves on four adjacent faces of the cube. Nine equally-spaced pinholes of 1 cm depth are located on the sides of the square groove on each face of the cube to accommodate steel pins. It has a central bore passing through the remaining 2

Fig 10: TGM[2] ISIS QA-1 Geometric QA Phantom

opposite faces of the cube. This central bore accommodates an object insert with a 2.54 cm Teflon sphere located in the center of the insert. It also has four built-in known electron density values of bone, water, and lung at inspiration and expiration.

Along with the ISIS QA phantom a leveling platform and alignment bar are used while acquiring the PET/CT dataset. ^{22}Na seeds are inserted in the pin holes for PET activity. A

Styrofoam block is used to hold the point sources required for measuring the resolution based on NEMA specification.

3.3 Software

In this session the software used for data acquisition and fusion, computation and RTP is discussed.

3.3.1 Acquisition and Fusion Software

The acquisition, reconstruction and fusion processing are all integrated together by the Windows NT-based Syngo platform software. This provides a common graphical user interface for PET and CT. The Syngo software is modality-independent software which supports PET, CT and even MRI images. Its components include the patient database browser, patient registration, 2D and 3D viewers, and DICOM archiving. The MultiSlice viewer system provides advanced multi-modality image display and data fusion capabilities. The Viewer can display a series of 4 image slices in a single display.

Significant Syngo image display tools include reorientation tools, region of interest tools and measurement tools. The reorientation tools include zoom, pan and rotate. Zoom and pan are used to adjust the display of tomographic slices by magnifying and moving the object. The PET and CT display can also be adjusted by thresholding the PET and CT windows. Usually the PET window threshold is set to define proper edges. The region of interest tools includes circle, ellipse and square shapes. Region of interest tools are used to measure the standard uptake value (SUV) or activity concentration. There are also measurement tools like the poly-line distance tool and the free-hand distance tool. These tools help to measure the dimension of an object based on pixel size.

3.3.2 Computation Software

PET/CT images are arrays of numerical values. These images are stored in DICOM format. These images should be analyzed by independent software for QA purposes. The analysis software used in this thesis is Mathematica 5. Mathematica loads the DICOM images into arrays for processing. Non-linear Gaussian curve fitting is used to find the centroid and resolution in the images.

3.3.3 RTP Software

The treatment planning software used to import and fuse PET/CT datasets is ADAC PINNACLE^3 treatment planning software. The treatment planning computer creates a 3D model of the tumor from the CT dataset. The planning software provides image fusion and virtual simulation tools. Patient image data are transferred to the system from the imaging devices. The system allows selecting specific patients or studies to be imported into the planning system. The system allows the operator to import and manually manipulate and fuse images from multiple modalities into composite images. In this work the PET and CT dataset are imported into the treatment planning software and geometric scaling accuracy, CT number accuracy and registration accuracy are verified.

CHAPTER 4
METHODS

4.1 Preparation of Radioactive Sources

In this session we will discuss how the ^{22}Na seeds and the capillary tube point sources are prepared.

4.1.1 Making of ^{22}Na Seeds

The ^{22}Na seeds used as sources for PET acquisition in ISIS QA-1 phantom were fabricated by physics faculty and a student of Louisiana State University as a part of another project [34]. The goal was to make ^{22}Na seeds of 5 mm length similar to dimensions of actual brachytherapy seeds, and that contain approximately 1 μci of activity inside. The ^{22}Na seeds are made from liquid ^{22}Na. Long cotton strands were threaded through the center of a surgical needle and a length of 5 cm was left hanging from the tip. The extra length was left to soak in the liquid ^{22}Na. After drying, the cotton thread was drawn up through the needle and pinched into 5 mm pieces using a wire cutter. The activities of the ^{22}Na seeds are tabulated in Table 2.

Table 2: ^{22}Na seed activities

^{22}Na seed activity	
Seed	Measured, μCi
1	0.35
2	0.22
3	0.46
4	0.51
5	0.49
6	0.25
7	0.4
8	0.36
9	0.55
Total	3.59

4.1.2 Preparation of Point Source

Performance of the imaging system is measured by the point spread function. Point sources have to be used to measure the performance of the imaging system as point sources are close to a delta function. Making a point source simulating delta function can be difficult tough task as it needs to be much smaller than the spatial resolution of the imaging system. The standard rule of thumb is that the source needs to be less than approximately one third of the FWHM spatial resolution. Capillary tubes of 1 mm inner diameter can be used to represent point sources. An FDG solution of 100 μCi/mL was prepared and pipetted into a centrifuge tube. About 1 μL of that was pipetted and dropped inside the tip of the capillary tube. Care is taken to make sure the extent of the activity does not increase beyond 1 mm, so the point source remains a delta function.

4.2 PET/CT Data Acquisition

In this session we discuss about setting up the phantom and choice of input and reconstruction parameters for PET/CT scan.

4.2.1 Phantom Setup

The TGM^2 ISIS QA-1 phantom was used to acquire PET/CT images for validation. Radioactive ^{22}Na seeds of approximately 5 mm length and 0.8 mm diameter were inserted in the pinholes of the cube. Before positioning the phantom on the patient table, the horizontal platform and the alignment bars were fitted onto the patient table of the PET/CT scanner. The horizontal platform was leveled by adjusting the leveling screws. The leveled platform is positioned exactly in the center of the table with the help of the alignment bar. The ISIS cube phantom with Teflon sphere object insert and ^{22}Na seeds in the pinholes was placed on the horizontal platform. Before starting the PET/CT scan, phantom alignment was checked by verifying that the laser lights align

with cross hair marks on the table alignment bar and on the laser grooves on the sides of the cube phantom. The patient table was moved up and down to have the object insert exactly at the isocenter (Fig 11).

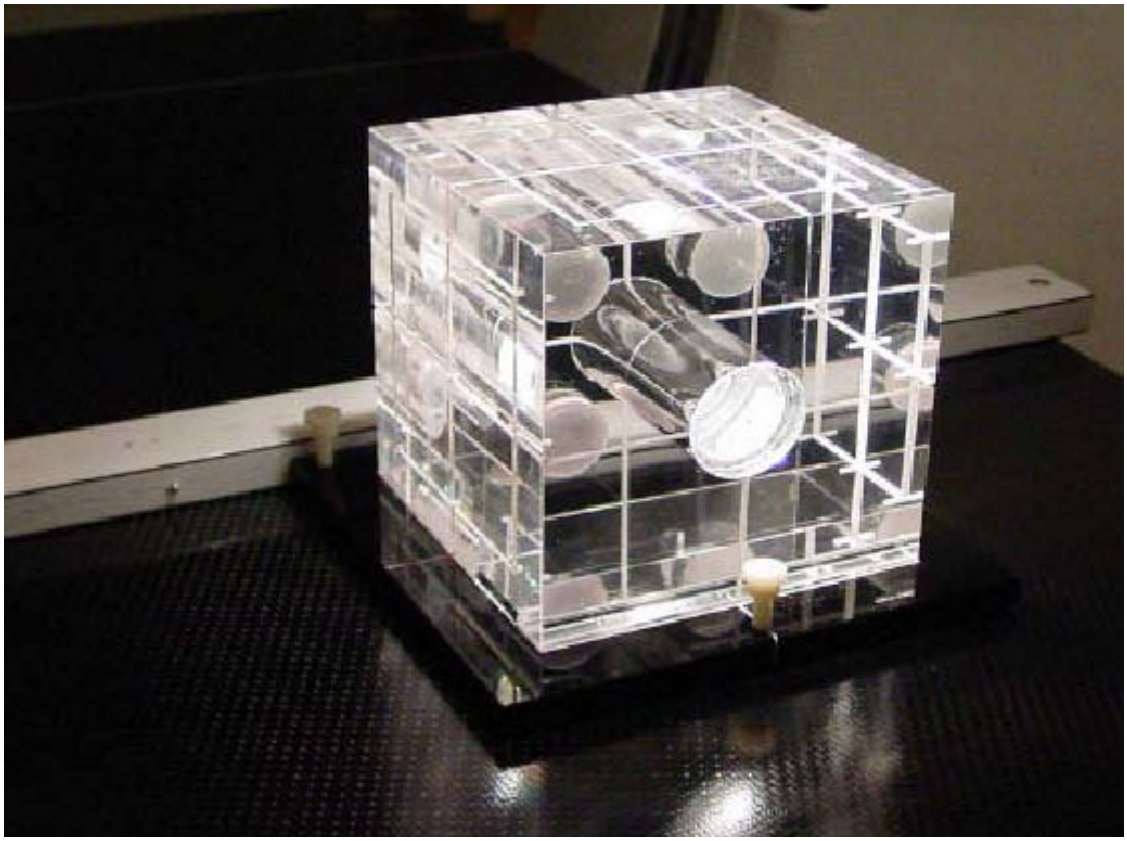


Fig 11: Set up of TGM^2 QA Phantom on flat patient table

4.2.2 Input and Reconstruction Parameters

Once the phantom is properly centered in the PET/CT scanner, the scanning is started. A patient registry is created, recording the phantom weight (3.6 Kg) and scanning protocol (thorax wholebody). After completing the phantom registration a topogram scan is acquired. The topogram displays the phantom on the patient table; scan range is chosen by spanning the axial FOV position markers above and below the phantom. The axial FOV is 15.5 cm and the 14 cm

phantom cube fits within the axial FOV (one bed position). The CT data were acquired for both 1 mm and 5 mm slice thickness. The other parameters such as scan time and mAs were chosen to be the minimum values. The Syngo software control has an optimization tool which only lets the scanning start for optimum parameters. The input parameters typically used for PET and CT images are given in Table 3.

Table 3: PET and CT acquisition parameters

PET and CT Acquisition Parameters		
CT		
	No of Bed	1
	KVP	130
	Eff mA	80
	Scan time	42 s
	Slice thickness	1, 5 mm
PET		
	Radio-isotope	^{22}Na
	Activity	3.59 µCi
	Injection time	hr:min:sec
	Scan time/bed	5 min

The different reconstruction parameters available for reconstruction are given in Table 4. Iterative reconstruction algorithms are typically used for clinical reconstruction. For resolution studies, filtered backprojection (FBP) algorithms are used. The Fourier Rebinning method (FORE) was used with iterative reconstruction and Single slice Rebining (SSRB) was used with FBP. A Gaussian axial filter of FWHM 3 mm was used for smoothing. Decay correction, dead time correction, attenuation correction and normalization are applied before reconstruction. Both attenuation corrected and non-attenuation corrected images are generated.

Table 4: Reconstruction parameters

Reconstruction Algorithm	3D Sinogram rebin method	Filter	Image size	Pixel size
Iterative	FORE	Gaussian	128x128	5.15 mm
FBP	SSRB	Axial filter	256x256	2.574 mm
DIFT	Segmentation	3mm FWHM	512x512	1.287mm

4.3 Data Analysis using Syngo Software

In this session we discuss how the Syngo software is used to measure the CT number values of electron density inserts, geometric scaling of pixels and visualization of PET and CT registration.

4.3.1 CT Number Analysis

CT images are nothing but a matrix of numbers representing density of the scanned object. These CT numbers give the attenuation of the object at that particular location. The performance of a CT scanner can be determined by measuring the CT values for a known electron density object. There are four electron density inserts (bone, water, inhale lung and exhale lung) in the TGM^2 phantom. The expected CT values for those electron density inserts as given by the manufacturer, are given in Table 5. Most CT QA protocols have a CT value check for daily QA. The Syngo region of interest tools were used to find the CT values (Fig 12). A circular region of interest is selected over each electron density insert in the CT image slice. The regions are selected to have equal area for all measurements. The measured CT values are compared with the manufacturer values to check the performance. The CT values are also checked in the RTP Software.

4.3.2 Geometric Scaling Analysis

The pixel size of the CT image matrix can be verified by measuring the dimensions of the scanned phantom. The CT images of the TGM^2 ISIS QA-1 phantom are loaded in the Syngo

Table 5: CT number value for electron density insert

	Electron density insert	CT value
1	Bone	800
2	Water	0
3	Exhale Lung	-500
4	Inhale Lung	-800

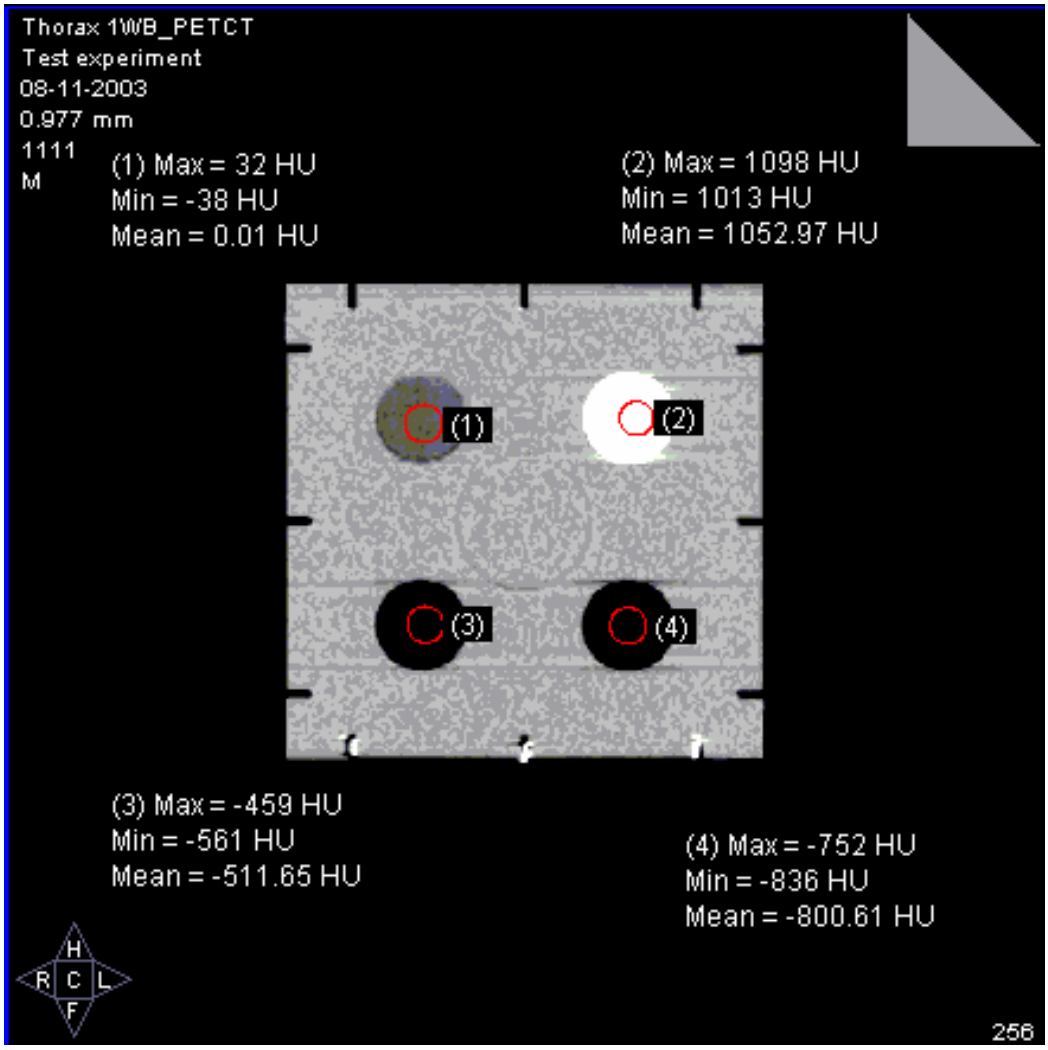


Fig 12: Syngo CT number measurement

MultiViewer system. The size of the phantom is measured in all three axial directions. The Syngo free hand distance measurement tool is used to measure the size of the phantom (Fig 13). The size of the Teflon ball is also measured. PET matrix translation is also done in a similar way by measuring the distance between two known points. The distance between two ^{22}Na seeds is measured with the free hand distance tool. User discretion is used to judge the center of the PET activity to determine the points from which the distance is measured. This measured distance is compared with the known distance. This distance translation is measured for a few randomly selected slices in the PET and CT images.

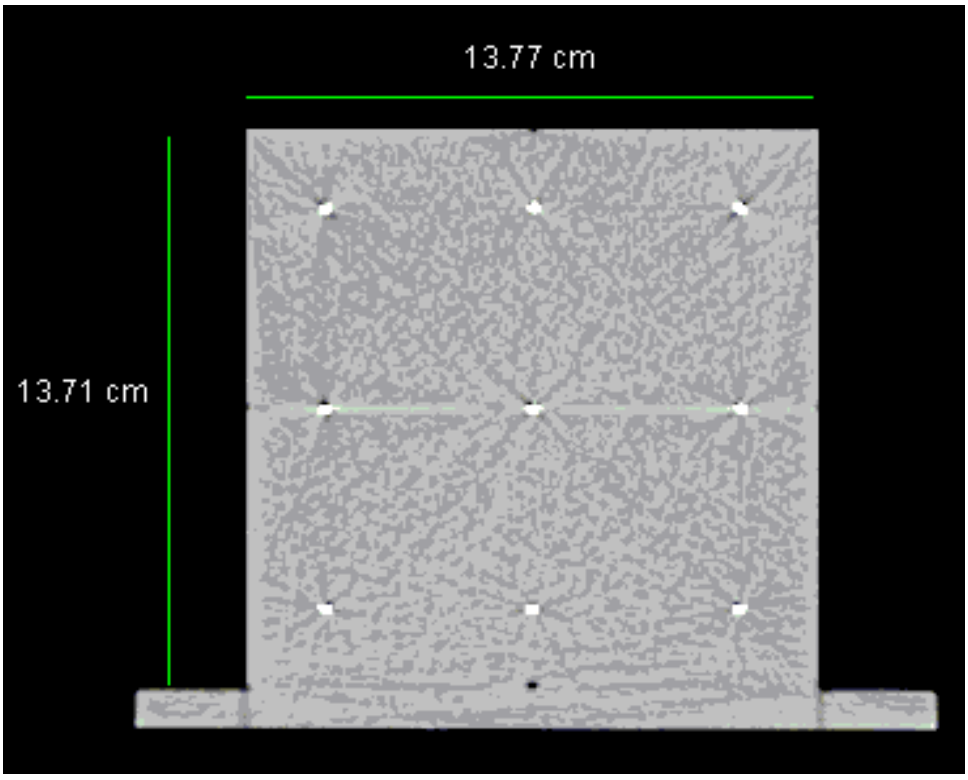


Fig 13: Syngo dimension analysis

4.3.3 Registration Error Analysis

Display of images is usually done by converting the digital value associated with each image pixel to a corresponding gray level display. A range of digital values are converted proportionally to shades of gray. The displayed gray scale values depend on the position of a window-level function on the digital values (Fig 14). The range of gray values is called the window, and the midpoint value of the range is called the level. The left edge of the window starts at the pixel value L-(W/2) and the right edge of the window is at pixel value L + (W/2). Where L is the level setting and W is the window setting. For normal image display where larger pixel values are displayed brighter, pixel values below (L – (W/2)) are truncated to 0 (black), and pixel values above L + (W/2) are saturated to white. This principle of windowing and leveling is used for both PET and CT images. By adjusting the window and level, the displayed range of values can be adjusted. For fused images, Syngo provides separate PET and CT window-level functions. These windows are adjusted to improve contrast. Also there is a PET/CT balance parameter to select the amount of PET and CT data visible in fused images.

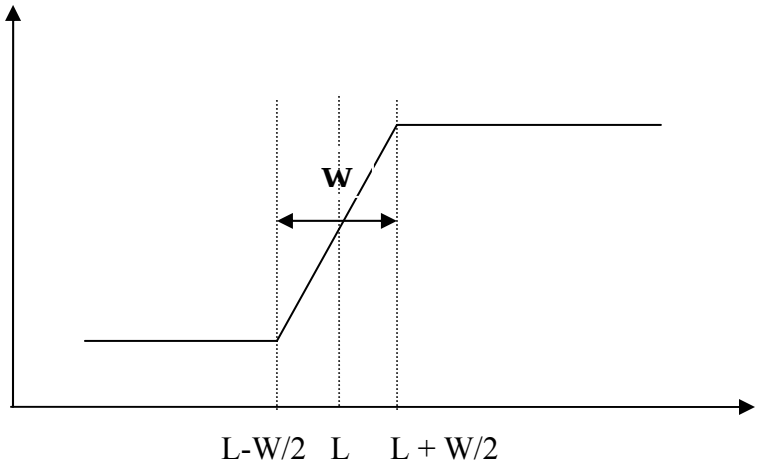


Fig 14: Widowing and leveling by adjusting contrast window

PET/CT image fusion is also done using the Syngo software. The CT images are loaded in the Multiseries viewer and PET images are loaded and fused in the 3D image fusion tool. The datasets are fused without any manual alignment. The fusion is checked by adjusting the window and level of PET and CT windows (Fig 15). The PET window is adjusted until only a single pixel is visible; the CT window is adjusted also. Then the PET and CT alignment is verified visually by checking the alignment of PET and CT points. The step by step process to check the fusion is outlined in Figure 16.

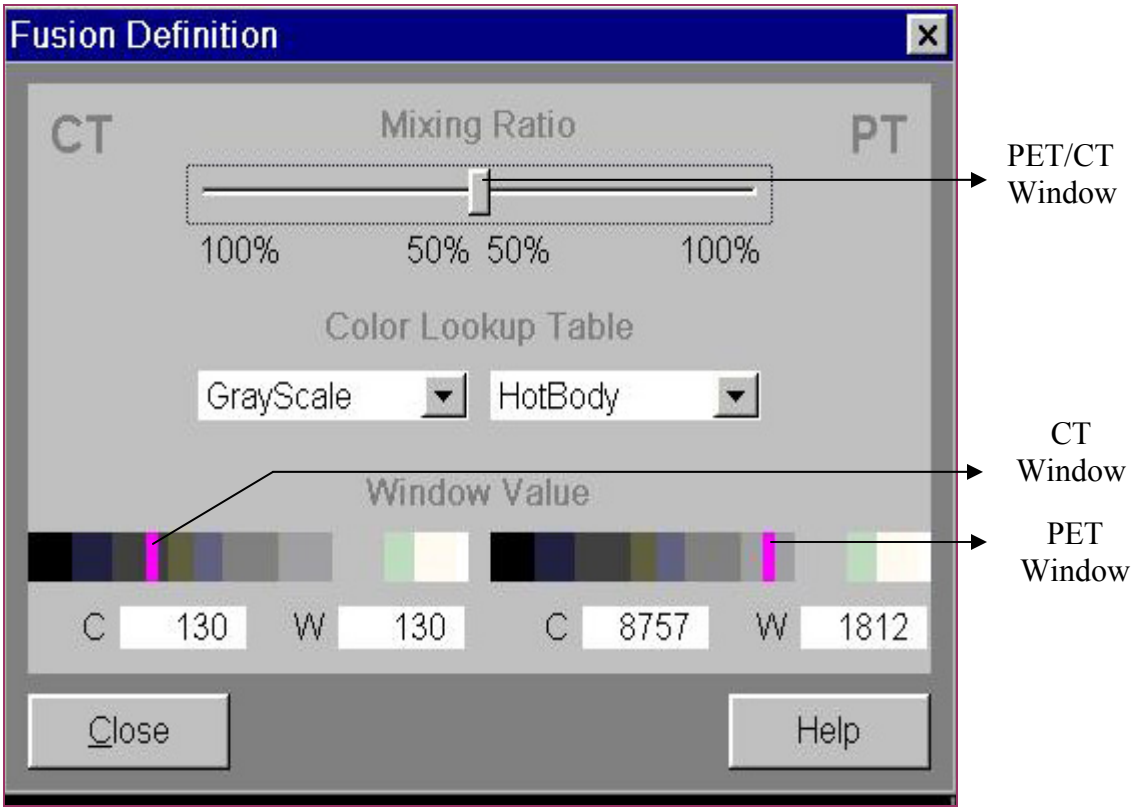


Fig 15: PET and CT windows

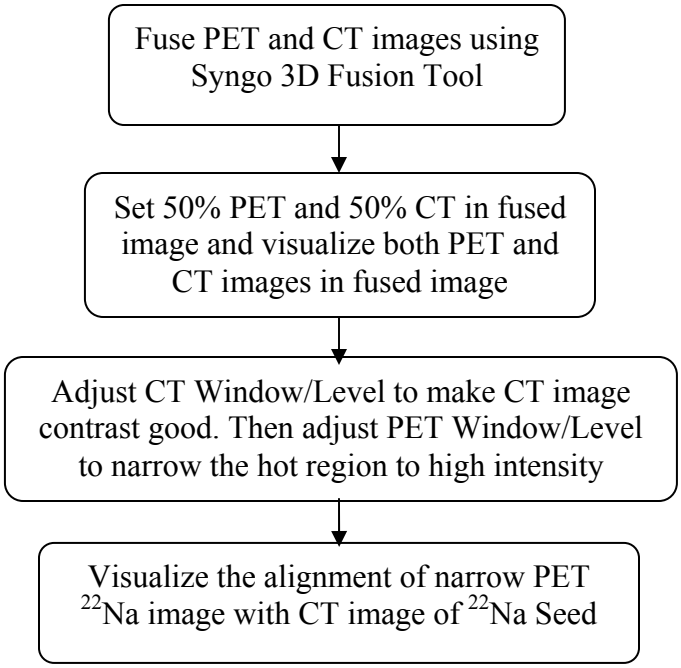


Fig 16: Flowchart for finding registration error using Syngo 3D fusion tool

4.4 Data Analysis in RTP System

The PET/CT dataset must be validated both in the scanner and in the RTP system. One of the inherent limitations in the RTP system is it can not import the fused PET/CT dataset. Although the PET/CT datasets are fused in the scanner workstation, they cannot be transferred as fused PET/CT images. Instead the CT images and PET images are send over the network as separate image series. The transferred CT and PET images are imported in the RTP system. The imported CT images are checked for their CT values, and matrix translation. The CT values are measured by region of interest tools in the RTP software and matrix translation is verified by measuring the dimensions of the phantom by software measurement tools. The dimension of the phantom is measured in the x, y and z directions. The CT values are checked for bone, water, and inhale and exhale lung.

After the CT dataset is loaded, the PET data is loaded to check image fusion. The Philips Pinnacle^3 ADAC treatment planning systems have the Syntegra auto fusion tool where the PET and CT dataset are fused automatically. The imported dataset is fused by using the auto-fusion tool in Syntegra. Both datasets once fused can be viewed in transverse, sagittal and coronal views. Any mis-registration can be found and measured using the measurement tools. Also the registration errors can be minimized by using the manual fusion tools available. In manual fusion, the PET dataset is dragged and fused by visually referring to CT landmarks.

4.5 Data Analysis using Mathematica Computation Software

In this session we discuss how we use mathematica to sum up the two dimensional image along the rows and columns, to curve fitting the data, determine the centroid, registration error analysis and geometric scaling accuracy of pixel.

4.5.1 Generation of One-Dimensional image plot from Two-Dimensional Image

Although the PET/CT fusion check and matrix translation check were done using the Syngo image tools, they should still be verified by independent software. Also, to find the quantitative resolution of the PET scanner, image analysis had to be done in software. Mathematic 5 developed by Mathworks is used for all image analysis work. The DICOM images are loaded into an array in Mathematica.

Analysis of images involves finding the pixel size for geometric scaling accuracy, finding the centroids from PET and CT regions for registration error analysis and counting the number of pixels between two points, and also finding the FWHM from PET regions by fitting nonlinear Gaussian curve. All these tasks start with the problem of locating the ^{22}Na seeds. The CT image of the phantom is usually a gray scale image having some value all over the region of the phantom. The seeds were located in the image by applying a threshold and converting the CT

image to a binary image. The metal seeds have a high CT number. By applying the threshold, the regions other than metallic seeds are turned to a black background (Fig 17). The PET image of the phantom (Fig 18) has image values only at the ^{22}Na seed location and all the rest of the region only has noise.

Both for PET and CT images one needs a one dimensional image plot for analysis. One dimensional image plots are used to find the center of CT and PET regions, for curve fitting analysis and for measuring pixel size. A one dimensional image plot is generated by summing up the pixels by row and by column (Fig 19). For PET images, the one dimensional plot gives a Gaussian-like distribution (Fig 20). This plot is fitted with a

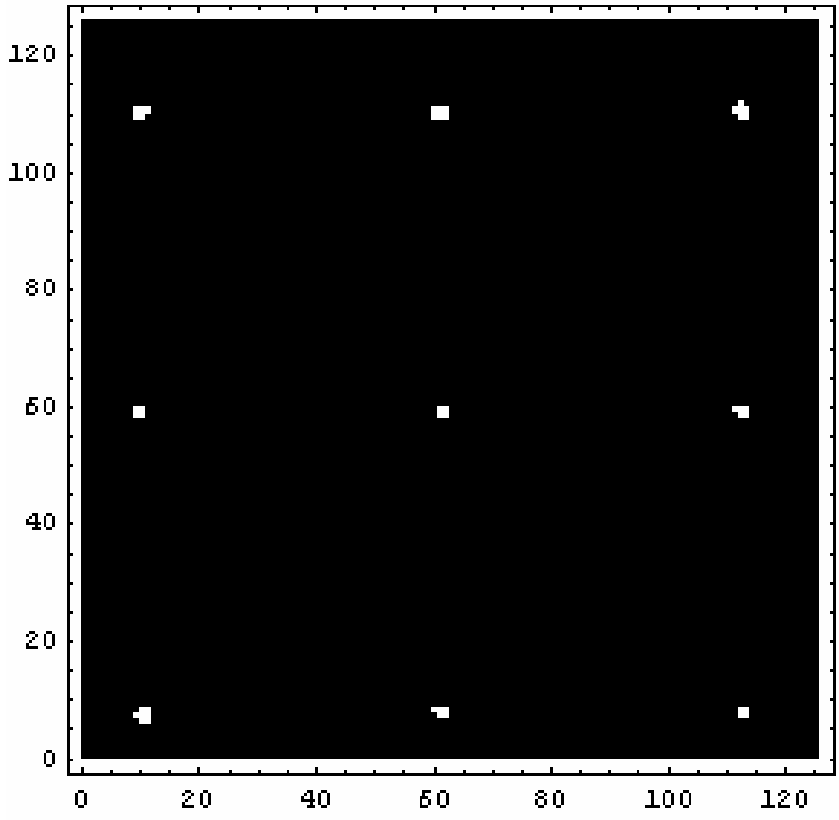


Fig 17: Binary CT phantom image

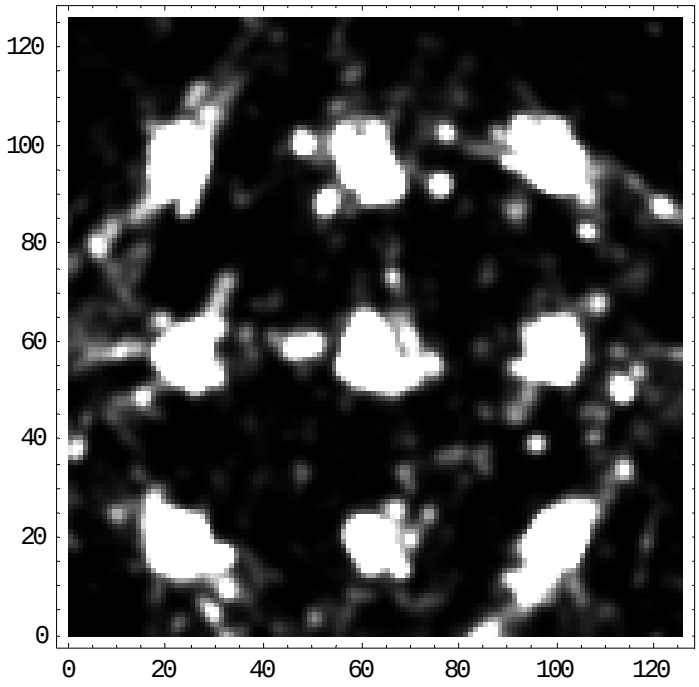


Fig 18: Cropped PET phantom image

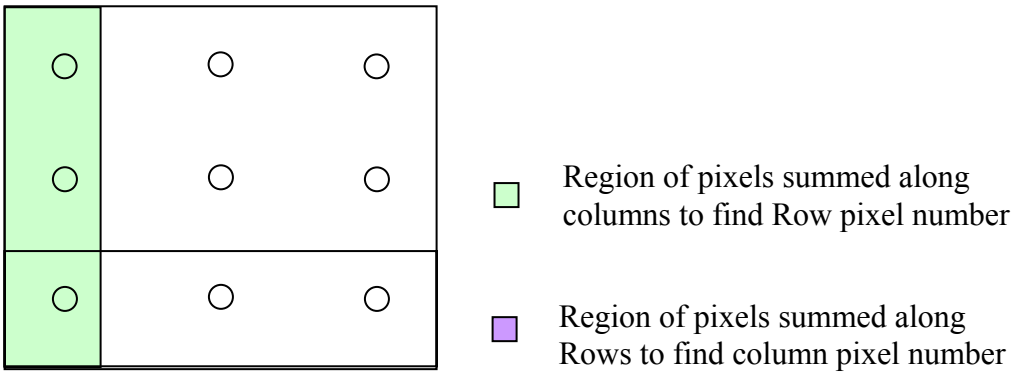


Fig 19: Summation of 2D gray scale image to 1D to find column and row sums

nonlinear Gaussian curve fitting techniques to find centroid and FWHM. For the CT image the one dimensional plot is used to find the pixel coordinate of white pixels. From the pixel coordinate the centers of the objects are found by finding the median pixel number.

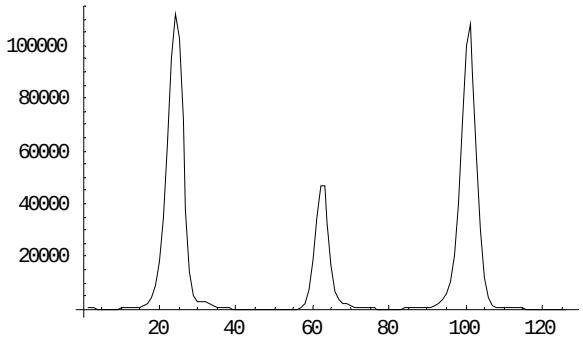


Fig 20: One dimensional PET image plot of the ^{22}Na seeds

4.5.2 Nonlinear Gaussian Curve Fitting Procedure

The goal of data fitting is to find the parameter values of a model which is assumed to closely match the data. To perform fitting, a function is defined which represents the closeness of the data to the model for particular parameter values. The function represents the difference between the data and the model dependent expected parameter values. This function is minimized to the smallest possible value with respect to the parameters. The parameter values that minimize the function are called the best fitting parameters [35]. A good understanding of the properties of the data is needed to choose the right model to represent the data.

The relationship between measurement value and measurement variable can be linear or nonlinear. Linear models can be fitted by linear regression equation and find the parameter values by solving the equation. Most models representing real life data are nonlinear. Nonlinear curve fitting also seeks to minimize the deviation between observed values and expected values by finding appropriate parameter values. In the nonlinear fitting process, usually iterative procedures are used instead of solving equations. Non-linear fitting usually starts with a guess of parameter values and finds the deviation from expected values. The deviation is minimized by changing the parameter values on successive iterations. Usually the user gives the initial

parameter value and number of iterations; the fitting software takes care of the rest of the iterative procedure.

Any analysis software can be used to find the best fitting parameter, but the model to represent the data need to be chosen by the user. In the case of PET images of ^{22}Na seeds, the ^{22}Na seeds are less than 1 mm in diameter and the pixel size of the PET matrix is usually greater than 1 mm. The images of the ^{22}Na seed activity distribution approximates the point spread function, i.e., the response of the imaging system to a delta function. Point spread functions are generally represented by a Gaussian model in PET imaging.

In the case with PET images nonlinear curve fitting techniques are used to find the FWHM of the point spread function of the PET images and also the center of the PET activity. Mathematica's nonlinear curve fitting functions are used to fit the Gaussian PET activity distribution. A Gaussian equation to represent the one dimensional plot in Fig. 20 is given in Fig. 21. Initial positions, initial background noise, and initial peak values are given as input for the iterative curve fitting function. The results generated give the calculated positions, background noise and standard deviation of the plot as shown in Fig. 22.

$$\text{answer} = \text{NonlinearRegress}\left[\text{data}, \text{amp1} \times \text{Exp}\left[\frac{-(x - \text{position1})^2}{2 \times \text{sigma1}^2}\right] + \text{amp2} \times \text{Exp}\left[\frac{-(x - \text{position2})^2}{2 \times \text{sigma2}^2}\right] + \text{amp3} \times \text{Exp}\left[\frac{-(x - \text{position3})^2}{2 \times \text{sigma3}^2}\right] + \text{Offset}, \{x\}, \{\{\text{amp1}, 100000\}, \{\text{position1}, 25\}, \{\text{sigma1}, 2\}, \{\text{amp2}, 40000\}, \{\text{position2}, 65\}, \{\text{sigma2}, 2\}, \{\text{amp3}, 100000\}, \{\text{position3}, 100\}, \{\text{sigma3}, 2\}, \{\text{Offset}, 100\}\}, \text{ShowProgress} \rightarrow \text{False}, \text{MaxIterations} \rightarrow 500\right]$$

Fig 21: Mathematica function for nonlinear Gaussian curve fit

	Estimate	Asymptotic SE	CI
amp1	111091.	702.116	{109701., 112482.}
position1	24.088	0.0145123	{24.0593, 24.1168}
sigma1	2.00085	0.0147797	{1.97158, 2.03013}
amp2	47861.8	750.176	{46375.9, 49347.6}
position2	62.4454	0.0315023	{62.383, 62.5078}
sigma2	1.75001	0.0320105	{1.68661, 1.81341}
amp3	106661.	700.358	{105274., 108049.}
position3	100.779	0.0151535	{100.749, 100.809}
sigma3	2.01103	0.0154341	{1.98046, 2.0416}
Offset	707.736	109.874	{490.116, 925.355}

Fig 22: Result of nonlinear Gaussian curve fit

4.5.3 Determination of Centroid in PET and CT images

For the case of verifying the registration of PET and CT images, the centroids of the PET and CT regions need to be found. Once the centroids of the regions of interest are found in both PET and CT images they can also be used to find the pixel size of the matrix. In the case of PET images finding the center of a PET region is a problem of fitting the Gaussian function to the activity distribution. The center can also be found by determining the pixel having the maximum value in the distribution. A PET distribution represents the two-dimensional distribution of activity in the x and y direction. To make the computation simple the two-dimensional function is transformed into a one-dimensional function as described before. The procedure of finding the centroid is given in Fig 24.

In the case of CT images the centroids are found from the binary CT phantom image. The binary image has ^{22}Na seeds as white spots with some noise. The pixels having white values are found and the center of that region of pixels is assumed to be the centroid. The procedure for finding the centroid in CT images is given in Fig 25.

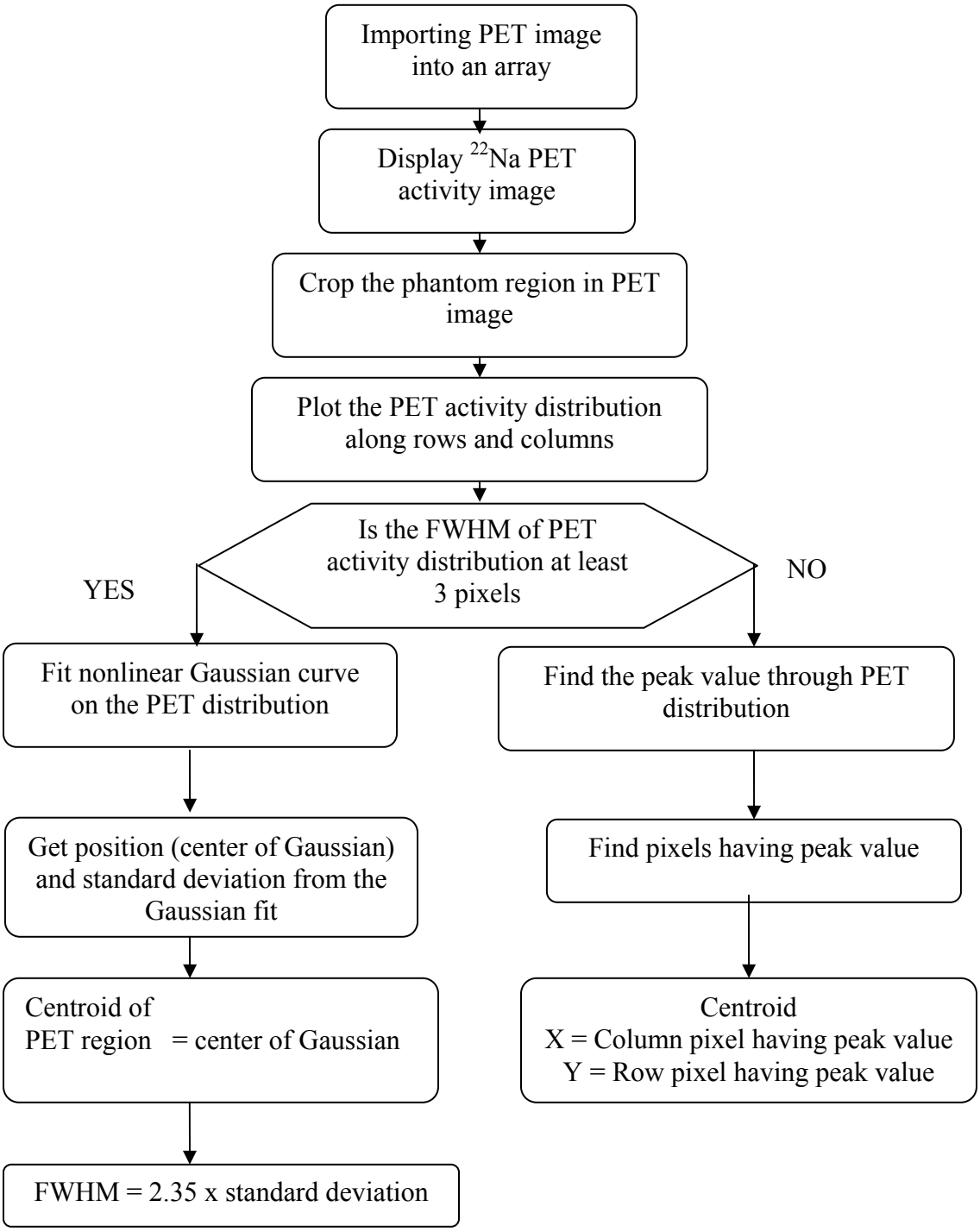


Fig 23: Flowchart of finding centroid and FWHM in PET image

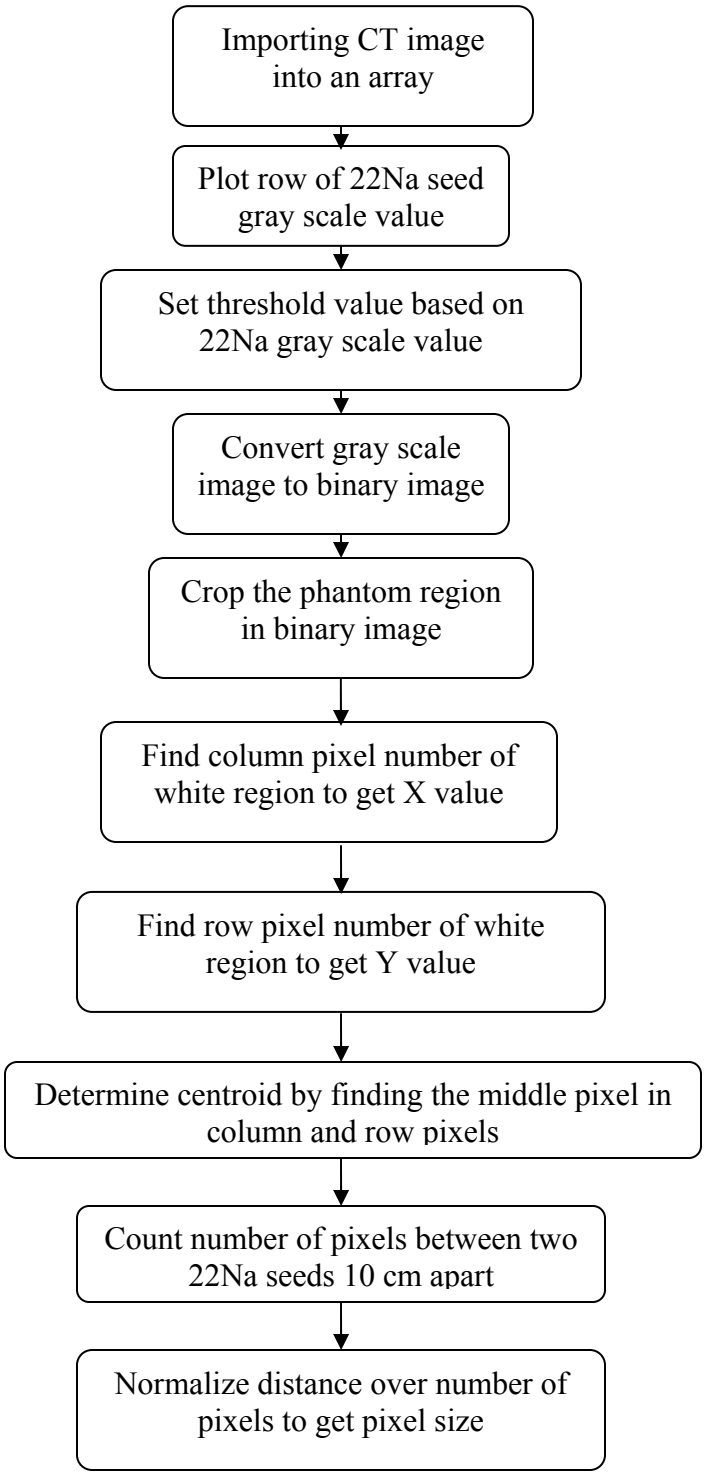


Fig 24: Flowchart to find centroid and pixel size in CT image

4.5.4 Registration Check for PET and CT Fused Images

PET/CT fused images are image matrixes of 512 x 512 arrays. The PET 128 x 128 arrays are stretched and fitted onto a 512 x 512 CT array. Also the field of view represented by the PET image is larger than the CT field of view. The fused PET/CT image now is a 512 x 512 array and the noise in the regular PET image is removed when it is fused with CT image. To check for registration in a PET/CT fused image, the PET/CT percentage window is adjusted sequentially for 100% PET and 100% CT to generate a PET-only image in the fused display and a CT-only image in the fused display. These image slices are saved as separate files to be analyzed for fusion. These images are again loaded into arrays and the centroid of the PET and CT regions are found as described before. Once the centroids of ^{22}Na seeds are found then the difference in the PET and CT centroids are computed to calculate the registration error. A flowchart of registration error analysis is given in Fig 26.

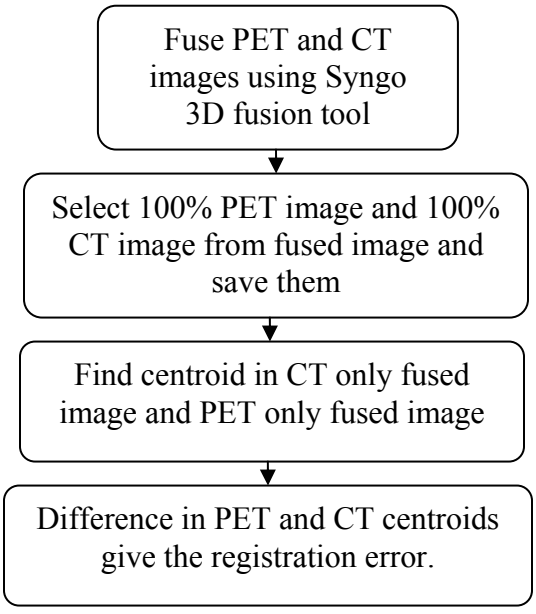


Fig 25: Flowchart of registration error analysis using Mathematica software

4.5.5 Geometric Scaling Accuracy Check

The size of pixels in the matrix defines the matrix translation to the dimension of the object. This pixel size is verified for CT images as well as PET images. In case of the CT phantom image, the pixel size is verified in x, y and z dimensions for 1 mm slice thickness and 5 mm slice thickness. In the CT phantom image, the center pixel of each ^{22}Na seed and pinhole located in the phantom are used as landmark points to calculate the pixel size. For the PET phantom image, the centroids of all ^{22}Na seeds are found. The number of pixels between the two 10 cm spaced ^{22}Na seeds or pinholes are calculated. The number of pixels spanning the 10 cm distance is used to calculate pixel size (Eq. 8). The pixel size is calculated for PET matrix, CT matrix and PET/CT fused matrix.

$$Pixel\,size = \frac{100\ mm}{Number\ of\ pixels} \qquad (8)$$

4.6 NEMA Resolution Analysis

The spatial resolution of a system represents its ability to distinguish between two points after image reconstruction. This measurement is performed by imaging point sources in air, and then reconstructing images with no additional smoothing [31, 32]. The measured spatial resolution by this means provides a best case comparison among scanners. The purpose of measuring the resolution is to characterize the width of the reconstructed image point spread function of point sources. The FWHM width of the spread function is measured by full width at half-maximum amplitude (FWHM).

To measure the width of the point spread function accurately the FWHM should span at least 3 pixels. The pixel size should be no more than one third of the expected FWHM in all three

dimensions during reconstruction. The pixel size of the PET image when reconstruction is done in a 512 x 512 matrix is 1.3 mm. This is the right matrix size for reconstruction to find FWHM as the expected FWHM is greater than three times the pixel size. The amount of activity for the point source is decided based on Poisson statistics to keep the noise level at a minimum.

A small quantity of concentrated activity (100 μCi/ml) fills the ends of glass capillary tubes. The extent of activity is 1 mm or less in all three dimensions. The sources were fixed parallel to the long axis of the tomography system and located at 6 points in a Styrofoam block. In the axial direction the sources were located in two planes with 3 sources at each plane. One plane is at the center of the axial FOV and other one is at one-fourth of the axial FOV. The axial planes where the sources are positioned are z = 0 cm and z = 4 cm. In each of these planes in the transverse direction the sources were positioned at (1) 1 cm vertically from the center, x = 0 cm and y = 1 cm (2) x = 0 cm and y = 10 cm, and (3) x = 10 cm and y = 0. The source arrangements are shown in the Figure 27.

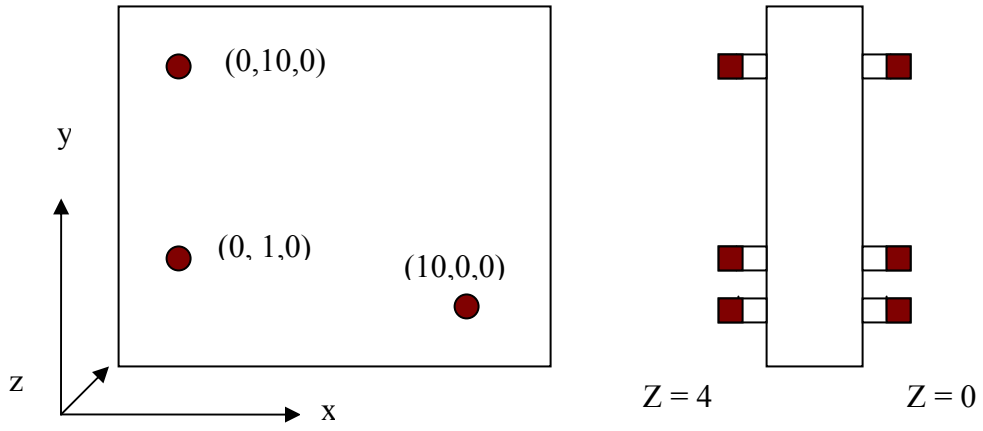


Fig 26: Styrofoam block inserted with FDG filled capillary tube point sources based on NEMA recommendation

The measurements were taken with all 6 sources. A minimum of one hundred thousand counts were acquired for each response function. The reconstruction was done by filtered

backprojection, rather than an Iterative algorithm with no smoothing filter. The point spread function in all three directions is determined by forming one dimensional response functions from two dimensional response functions as described before. The FWHM can be determined by counting the number of pixels at half the maximum value of the response function. This value in pixels will be converted to distance in millimeters by multiplication by pixel size. Then the FWHM values computed for each point source are used to calculate the resolution based on NEMA formulas. The flowchart of resolution analysis procedures is given in Fig 27 and the NEMA formulas are given in Table 6.

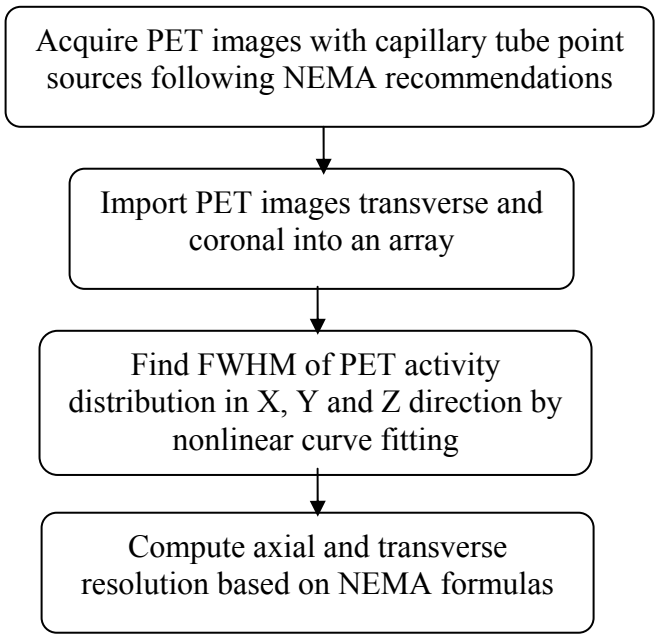


Fig 27: Flowchart of computing resolution based on NEMA recommendation

Table 6: Formulae for computing spatial resolution report values

	Description	Formula
At 1 cm radius		
Transverse	Average x & y for both z positions (4 numbers)	$RES = \{ RESx_{x=0,y=1,z=0} + RESy_{x=0,y=1,z=0} + RESx_{x=0,y=1,z=4} + RESy_{x=0,y=1,z=4} \}/4$
Axial	Average of 2 z positions (2 numbers)	$RES = \{ RESz_{x=0,y=1,z=0} + RESz_{x=0,y=1,z=4} \}/2$
At 10 cm radius		
Transverse radial	Average 2 transverse for both z positions (4 numbers)	$RES = \{ RESx_{x=0,y=1,z=0} + RESy_{x=0,y=1,z=0} + RESx_{x=0,y=1,z=4} + RESy_{x=0,y=10,z=4} \}/4$
Transverse tangential	Average 2 transverse for both z positions (4 numbers)	$RES = \{ RESx_{x=0,y=10,z=0} + RESy_{x=10,y=0,z=0} + RESx_{x=0,y=10,z=4} + RESy_{x=10,y=0,z=4} \}/4$
Axial resolution	Average 2 transverse for both z positions (4 numbers)	$RES = \{ RESz_{x=10,y=0,z=0} + RESz_{x=0,y=10,z=0} + RESz_{x=10,y=0,z=4} + RESz_{x=0,y=10,z=4} \}/4$

(RESx, RESy, RESz refer to spatial resolution in the x, y, and z- directions)

CHAPTER 5
RESULTS AND DISCUSSION

5.1 CT Number Analysis

The performance of the CT scanner is verified by checking the CT number of the four electron density inserts bone, water, inhale lung and exhale lung in TGM[2] ISIS QA-1 phantom. The CT numbers were measured using Syngo software as described in Chapter 4. These measured Hounsfield unit of electron density inserts were compared with the expected values given by the manufacturers. The measured CT number values for the inserts are given in the Table 6 for 5 mm and Table 7 for 1 mm slice thickness. The CT number of the bone was found to be 28% offset. The errors in CT numbers of other inserts were marginal.

Table 7: Measured CT number values for 5 mm slice thickness

Measured CT Number in PET/CT Scanner				
Measurement	Bone (800 HU)	Water (0 HU)	ExLung (-500 HU)	Inhale (-800 HU)
1	1027	4	-507	-794
2	1030	4	-508	-804
3	1030	4	-507	-806
4	1036	6	-504	-801
Mean	1030.8	4.5	-506.5	-801.3
SD	3.78	0.78	1.9	5.3
Error	230.8025	4.42	-6.4225	-1.655

Table 8: Measured CT number values for 1 mm slice thickness

Measured CT Number in PET/CT Scanner				
Measurement	Bone (800 HU)	Water (0 HU)	ExLung (-500 HU)	Inhale (-800 HU)
1	1048	3	-495	-794
2	1051	0	-498	-787
3	1056	1	-507	-757
4	1023	0	-500	-798
Mean	1044.5	1	-500	-784
SD	14.51	1.23	5.13	18.72
Error	244.62	1.18	0	16

The CT number values were also checked in CT images imported in the RTP workstation. The measured CT number values are tabulated in Table 9 and Table 10 for 5 mm and 1 mm slice thickness. The CT number measured in the RTP software was offset for all the electron density inserts. They were offset by a constant value. Based on the observed pattern of increase in measured CT number values the increase is found to be approximately 1000. The likely reason for the shift of CT number in RTP software is the software may scale HU from 0 to 4000 instead of the usual -1000 to 4000. In that case the

$$\text{CT Number (HU)} = 1000\frac{\mu_{pixel}}{\mu_{water}} \qquad (9)$$

definition of CT number is given in Eq. 9. The dose for each pixel need to be calculated based on attenuation derived from Eq. 9.

Table 9: Measured CT number value in RTP system for 5 mm slice thickness

Measured CT Number in RTP system				
Measurement	Bone (800 HU)	Water (0 HU)	ExLung (-500 HU)	Inhale (-800 HU)
1	2049	1032	508	218
2	2053	1015	498	216
3	2044	1022	520	231
4	2000	1037	545	266
Mean	2036.5	1026.5	517.75	232.75
SD	24.61	9.883	20.271	23.143
Error	1236.5	1026.5	1017.75	1032.75

5.2 Geometric Scaling Analysis

The CT images of the TGM2 ISIS QA-1 phantom are checked for geometric scaling error in the scanner by measuring the dimension of the phantom in x, y and z directions using Syngo software as described in Chapter 4. The CT images of the TGM2 phantom were acquired for slice thicknesses of 5 mm and 1 mm. The measured dimensions of the phantom are given in

Table 10: Measured CT number value in RTP system for 1 mm slice thickness

Measured CT Number in RTP system				
Measurement	Bone (800 HU)	Water (0 HU)	ExLung (-500 HU)	Inhale (-800 HU)
1	2062	1043	530	241
2	2053	1047	534	239
3	2057	1043	542	238
4	2045	1039	525	237
Mean	2054.25	1043	532.75	238.75
SD	7.182	3.266	7.182	1.708
Error	1254.25	1043	1032.75	1038.75

Table 11 and Table 12. The measured dimension in Z direction was found to be -5.2% offset for 5 mm slice thickness and -2.5% offset for 1 mm slice thickness. One of the reasons for scaling inaccuracy in the axial direction is the helical interpolation used to acquire the CT data of the phantom. This results in the edges of the phantom being approximated between air and acrylic. The distance between the ^{22}Na seeds were also measured. The measured distances are tabulated in Table 13.

The acquired CT images are transferred to the RTP Workstation and the geometric scaling of the phantom is verified again in x, y and z direction in the RTP system. The measured dimension is given in Table 14 and Table 15 for slice thickness of 5 mm and 1 mm. The scaling accuracy was found to be better in the RTP workstation. But still the scaling in z direction was inaccurate for a slice thickness 5 mm. The dimension of the phantom in z direction was -2.9% offset for 5 mm slice thickness. The errors were insignificant for 1 mm slice thickness. Geometric scaling error of the matrix was also verified by measuring the pixel size of the matrix. The pixel size is calculated for CT, PET and PET/CT fused matrixes. The pixel size of the CT matrix is found by

Table11: Measured cube dimensions for 5 mm CT slice thickness

Measured size of cube in PET/CT Scanner			
Measurement	X , cm	Y, cm	Z, cm
1	13.87	13.89	13.23
2	13.91	13.93	13.16
3	13.74	13.85	13.25
4	13.83	13.83	13.23
5	13.74	13.85	13.08
Mean	13.818	13.87	13.19
SD	0.077	0.04	0.070
Error	-0.182	-0.13	-0.81

Table 12: Measured cube dimensions for 1 mm CT slice thickness

Measured size of cube in PET/CT Scanner			
Measurement	X, cm	Y, cm	Z, cm
1	13.84	13.96	13.62
2	13.89	13.84	13.55
3	13.96	13.89	13.73
4	13.84	13.86	13.62
5	13.96	13.96	13.73
Mean	13.898	13.902	13.65
SD	0.060	0.056	0.078
Error	-0.102	-0.098	-0.35

Table 13: Measured distance in PET image between ^{22}Na seeds 10 cm apart

Measurement	X, cm	Y, cm
1	9.83	10.04
2	9.96	10.29
3	10.03	9.96
4	10.14	10.02
5	9.96	9.89
6	10.16	10.12
Mean	10.013	10.053

Table 14: Measured cube dimensions for 5 mm CT slice thickness in RTP system

Measured size of cube in RTP System			
Measurement	X, cm	Y, cm	Z, cm
1	13.87	13.831	13.663
2	13.804	13.961	13.599
3	13.739	13.896	13.663
4	13.831	13.961	13.534
5	13.896	13.896	13.469
Mean	13.828	13.909	13.586
SD	0.061	0.054	0.084
Error	-0.172	-0.091	-0.414

Table 15: Measured cube dimensions for 1 mm CT slice thickness in RTP system

Measured size of cube in RTP System			
Measurement	X, cm	Y, cm	Z, cm
1	13.961	14.09	13.987
2	14.026	13.961	13.922
3	14.091	13.896	13.955
4	13.961	14.091	13.987
5	14	14.026	13.922
Mean	14.008	14.013	13.955
SD	0.054	0.085	0.033
Error	0.008	0.013	-0.045

locating the ^{22}Na seeds in the TGM^2 phantom CT image. The ^{22}Na seed pixels are located in the CT image and the centroid is calculated as described in Chapter 4. Once the centroid of the ^{22}Na seeds is found, the number of pixels between two ^{22}Na seeds separated 10 cm apart is found. The calculated number of pixels for CT slice thickness of 5 mm and 1 mm are given in Table 16 and Table 17. Then the pixel size is calculated by equating the distance and the number of pixels. The calculated pixel size based on CT field of view and the array size is 0.977 mm. The measured pixel size in x, y and z directions are given below in Table 18 and Table 19.

Table 16: Number of pixels for 10 cm distance for CT slice thickness of 5 mm

No.	X pixels	Y pixels	Z pixels
1	103	102	102.5
2	103	102	103
3	103	102	103.5
4	103	102	103
5	103	102	101.5
Mean	103	102	102.7
Sd	0	0	0.758

Table 17: Number of pixels for 10 cm distance for CT slice thickness of 1 mm

No.	X pixels	Y pixels	Z pixels
1	102.25	102.5	102
2	102.75	103	102.5
3	102.25	102.5	102.35
4	102.5	102.5	102.5
5	102.5	103	102
6	102.5	102	102.25
7	103	102.5	102
8	102.5	102	102
9	103	102.5	102
10	102.5	102.5	102.5
Mean	102.575	102.5	102.21
sd	0.265	0.333	0.234

Table 18: Measured pixel size for CT matrix of 5 mm slice thickness

Distance = 100 mm	No of pixels	Standard deviation	Pixel size,mm	Standard deviation,mm
X	103	0	0.970	0
Y	102	0	0.980	0
Z	102.7	0.758	0.973	0.007

Table 19: Measured pixel size for CT matrix of 1 mm slice thickness

Distance = 100 mm	No. of pixels	Standard deviation	Pixel size, mm	Standard deviation, mm
X	102.575	0.265	0.975	0.003
Y	102.5	0.333	0.976	0.003
Z	102.21	0.234	0.978	0.002

The pixel size of the PET matrix is determined by finding the centroid of the PET activity region. The centroids of ^{22}Na seeds in the PET image are computed as described in Chapter 4. The typical PET image is reconstructed in a 128 x 128 matrix. Once the centroid is found the number of pixels between the two ^{22}Na seeds spaced 10 cm apart is counted. The number of pixels measured is given in Table 20. The number of pixels equated with the distance gives the measured pixel size. The measured pixel size is tabulated in Table 21.The specified pixel size by the Syngo software for a 128 x 128 matrix is 5.14 mm. Comparing the measured pixel size with the Syngo pixel size, the pixel size in y direction was offset by 2%.

The pixel size is also measured for the fused PET/CT matrix. The FOV of PET and CT is different. The PET FOV is larger than the CT FOV. In the PET/CT fused image, the 128 x 128 PET matrix with higher FOV is fused with the 512 x 512 CT matrix and the fused image matrix is again 512 x 512 and representing the area of the PET FOV. The pixel size of CT is expected to increase because FOV increases and the pixel size of PET will decrease as 128 x 128 matrix data is stretched to 512 x 512. The number of pixels for 10 cm distance is calculated based on centroids of PET and CT parts of the fused image. This centroid is also used for registration error check. The measured number of pixels for PET and CT images is given in Table 22 and Table 23. The measured pixel size is given in Table 24. The calculated centroid for PET only and CT only fused images are tabulated in Table 25 and Table 26.

Table 20: Number of pixels for 10 cm distance in PET image

No.	X, pixels	Y, pixels
1	20	19
2	19.5	19
3	19.5	19
4	19	19
5	19.5	19
6	19.5	19
7	19	19.5
8	19.5	19
9	19.5	19
10	19	19.5
11	19.5	19
12	19.5	19
Mean	19.417	19.083
Sd	0.289	0.195

Table 21: Measured pixel size for PET matrix

Distance = 100 mm	No of pixels	Standard deviation	Pixel size, mm	Standard deviation, mm
X	19.417	0.289	5.150	0.077
Y	19.083	0.195	5.240	0.053

Table 22: Number of pixels in PET only fused image for 10 cm

Measurement	X, pixels	Y, pixels
1	76.85	76.656
2	77.048	76.591
3	76.237	77.064
4	76.748	76.419
5	76.911	76.248
6	76.128	76.707
Average	76.654	76.614
STD	0.379	0.278

Table 23: Number of pixels in CT only fused image for 10 cm

Measurement	X, pixels	Y, pixels
1	77.5	78
2	78.5	77
3	78	77.5
4	78	77.5
5	78	77
6	78	77.5
Average	78	77.417
STD	0.316	0.376

Table 24: Measured pixel size for PET and CT only fused image matrixes

Distance = 100 mm	No. of pixels	Standard deviation, mm	Pixel size, mm	Standard deviation, mm
PET Matrix				
X	76.654	0.379	1.305	0.007
y	76.614	0.278	1.305	0.005
CT Matrix				
X	78	0.316	1.282	0.005
Y	77.417	0.376	1.292	0.006

5.3 Registration Error Analysis

The alignment of the PET and CT images of the TGM^2 Phantom with ^{22}Na seeds are checked in the registration error analysis. Registration check is done in Syngo software as well as by Mathematica programs. In Syngo the alignment of the PET and CT images can be visualized by adjusting the contrast windows. As example contrast-adjusted PET and CT fused image is given in Figure 28. The quantitative registration error is calculated by computing the centroid from Mathematica programs. The centroid for PET and CT only fused images are shown in Table 25 and Table 26. The registration error is tabulated in Table 27. The registration error is found to be approximately 1 mm in the scanner workstation.

Table 25: Computed centroid pixels in PET only fused image

	Measured Centroid values in PET Image					
	Column pixels, X value			Row pixels, Y values		
Slice1						
	218.588	257.219	295.438	215.851	215.872	216.572
	217.611	257.124	294.659	255.047	255.51	255.525
	218.588	257.219	295.438	292.507	292.463	293.636
Slice2						
	217.594	256.763	293.722	215.965	216.182	216.926
	218.547	257.143	295.295	254.874	255.394	255.517
	217.671	257.035	294.582	292.384	292.43	293.633

Table 26: Computed centroid pixels in CT only fused image

	Measured Centroid values in CT Image					
	Column pixels, X value			Row pixels, Y values		
Slice1	217	256	295	215	215.5	215.5
	217	256	295.5	254.5	254.5	254.5
	217.5	256.5	295	293	292.5	293
Slice2	217	256	295	215	216	215.5
	217	256.5	295	254.5	254	254.5
	217.5	256	295.5	292.5	293	293

Fusion in the RTP system also needs to be verified as the fused images cannot be send to the RTP system directly. The PET and CT images are imported separately and fused using Syntegra. The fusion done by using auto fusion tools were offset and they needed manual adjustment. The misalignment between the PET and CT datasets was measured and tabulated in Table 30. The fused PET/CT image is shown in Figure 29 and Figure 30. The misalignment was found to be approximately 6 mm.

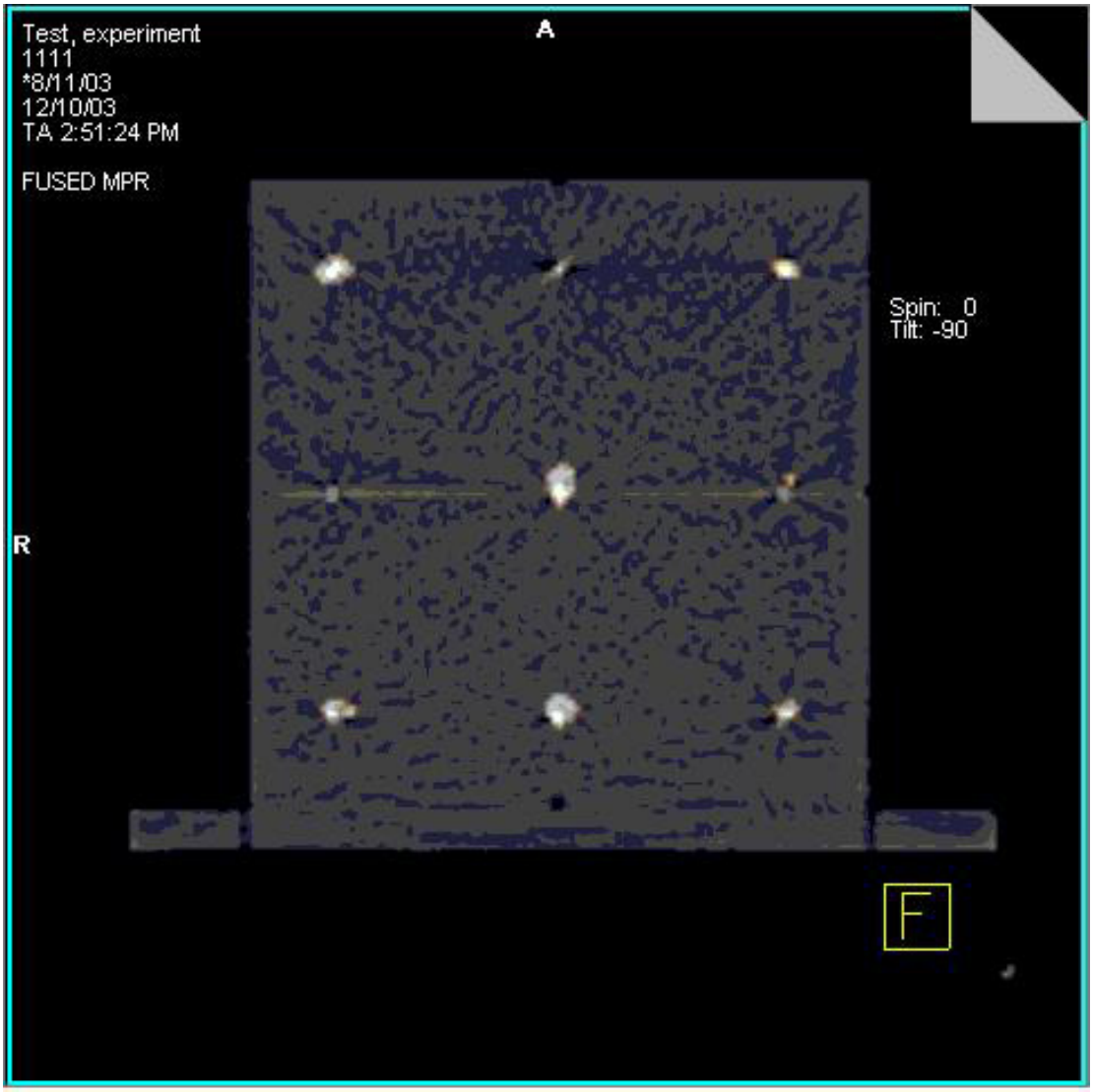


Fig 28: Syngo PET/CT fused image

Table 27: Computed registration error in pixels in X and Y direction

Registration error in X and Y axis in pixels for fused image		
Number	X, pixels	Y, pixels
1	1.088	1.047
2	0.611	0.671
3	0.574	0.594
4	0.719	1.143
5	1.124	0.535
6	0.692	0.763
7	0.438	0.205
8	0.841	0.418
9	1.189	1.278
10	1.047	0.965
11	0.671	0.182
12	0.594	1.426
13	1.143	0.374
14	0.535	1.394
15	0.763	1.017
16	0.205	0.116
17	0.418	0.57
18	1.278	0.633
Mean	0.774	0.741
sd	0.307	0.413

Table 28: Computed registration error in PET/CT fused image in pixels and mm

	Registration error, Pixels	Standard deviation	Pixel size, mm	Registration error, mm	Standard deviation
X	0.774	0.307	1.293	1.000	0.397
Y	0.741	0.413	1.299	0.962	0.537

Table 29: Measured FWHM in pixels at different points based on NEMA 2 – 2001

Measured FWHM in pixels at different point based on NEMA 2 – 2001							
No	(X,Y,Z)	Resx	stdev	Resy	Stdev	Resz	stdev
1	(10,0,0)	5.538	0.221	4.697	0.228	6.013	0.075
2	(0,1,0)	3.678	0.150	3.475	0.141	4.439	0.196
3	(0,10,0)	3.982	0.349	4.628	0.093	5.313	0.078
4	(10,0,4)	5.559	0.226	4.734	0.277	5.742	0.099
5	(0,1,4)	3.724	0.151	3.490	0.136	4.544	0.109
6	(0,10,4)	4.022	0.337	4.673	0.010	5.348	0.087

Table 30: Measured registration error of PET/CT dataset in RTP system

No	Registration Error, cm
1	0.638
2	0.567
3	0.707
4	0.658
5	0.486
6	0.638
7	0.567
8	0.612
9	0.452
Mean	0.592
Sd	0.082

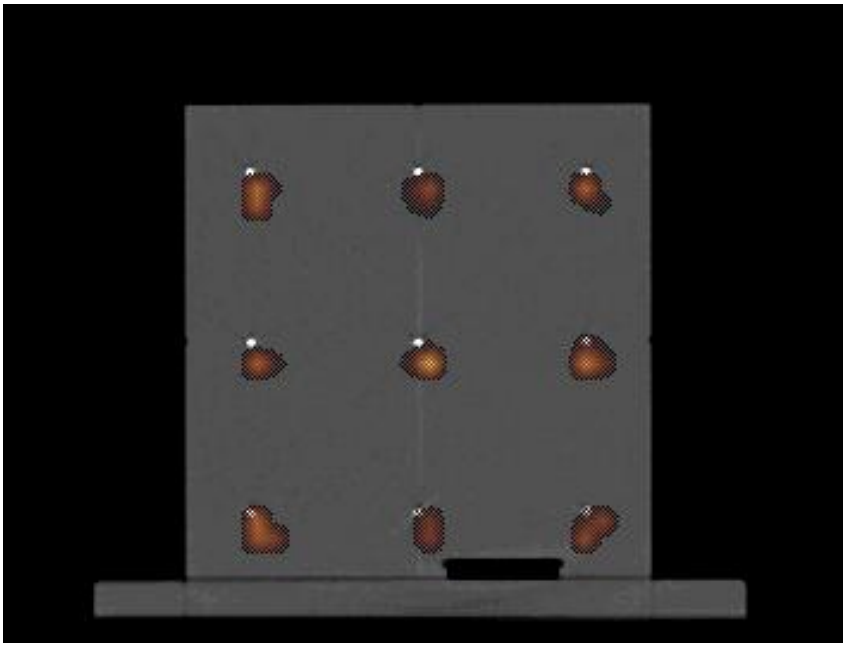


Fig 29: PET/CT fused image in RTP system using Syntegra

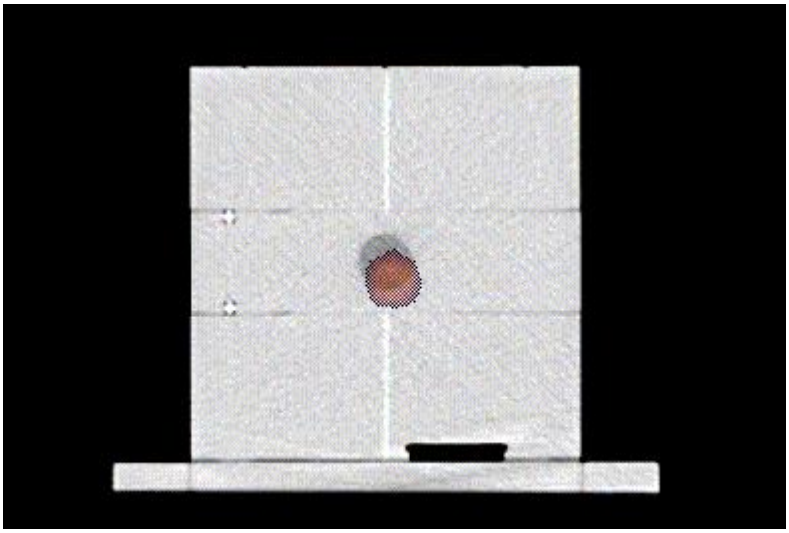


Fig 30: Fused image of TGM^2 phantom with PET object insert in RTP system

5.4 Resolution Analysis

As part of the validation of the PET/CT dataset the resolution of the PET scanner is also verified. The resolution measurement was done as described in Chapter 4. To find the resolution the FWHM of the point sources were calculated. The measured FWHM is displayed in Table 30. The resolution is calculated from the FWHM by NEMA formulas. The calculated resolution is given in Table 31. A comparison of measured resolution with the manufacturer's specified resolution is given in Table 32.

Table 31: Measured resolution based on NEMA 2001

512 x 512 Matrix	Resolution		Resolution	
Pixel size = 1.287 mm	pixels	Stdev	mm	Stdev
At 1 cm radius				
Transverse Resolution	3.592	0.072	4.623	0.093
Axial resolution	4.491	0.112	5.781	0.144
At 10 cm radius				
Transverse radial resolution	5.099	0.086	6.564	0.111
Transverse tangential resolution	4.359	0.151	5.610	0.194
Axial resolution	5.604	0.043	7.213	0.055
Transverse resolution	4.729	0.087	6.087	0.112

Table 32: Comparison of specified resolution and measured resolution

	Resolution specified value, mm	Resolution measured value, mm
At 1 cm radius		
Transverse Resolution	4.5	4.6
Axial resolution	4.2	5.8
At 10 cm radius		
Transverse Resolution	5.6	6.1
Axial resolution	5.7	7.2

CHAPTER 6
CONCLUSION

The TGM^2 phantom was helpful in analyzing the geometric scaling, CT number and PET/CT registration of the PET/CT scanner. The geometric scaling accuracy, CT number accuracy and PET/CT registration accuracy were verified in the scanner software as well as in the RTP system software. The CT numbers of the different electron density inserts were measured and compared with the TGM^2 specified values. The dimension of the electron density insert in the TGM^2 phantom is 2.5 cm diameter and 1.5 cm thick.

The measured CT number value of the bone density insert was found to be incorrect in both the 5 mm and 1 mm CT slices. The measured CT number values are based on the densities and since the CT slice thickness used in the QA checks are 5 mm and 1 mm, there is no issue of partial volume effect as the electron density inserts are thicker than the CT slice thickness. Also the TGM^2 specified values were obtained for a clinical range of parameters, i.e. 120 kVp and 200 mAs, for 10 mm CT slice thickness. The CT scanning parameters used in this QA scan, i.e. 130 kVp and 80 mAs, are close to the TGM^2 specified parameters. The kVp difference is minimal and the mAs difference will only change the noise level in CT values. The measured CT values using these scan parameters for all other density inserts were found to be correct except for the bone density insert. Hence the mis-measurement of bone density insert should be reported to a field service engineer. Interestingly this mis-measurement of bone density insert went unnoticed by regular daily QA checks as the regular daily QA phantom does not have a bone density insert.

The CT numbers of the electron density inserts were again measured in the RTP software after importing the CT images into the RTP system. The CT numbers were found to be shifted from the scanner measured values by a constant value. This shift in the CT number values is due to a different CT number scaling in the RTP software to represent physical density. These scaled

CT values used in dose optimization algorithms help in avoiding the use of negative and zero values for physical density.

The geometric scaling accuracy in the Syngo scanner software and the RTP software was verified by measuring the dimensions of the TGM^2 phantom. The phantom dimensions were measured in x, y and z directions. The geometric scaling accuracy was found to be inaccurate along the z axis. The scaling error in the z axis is due to helical interpolation during reconstruction and partial volume effect. Not surprisingly the scaling accuracy improved with thinner slices by limiting the partial volume effect in the edge of the phantom. Interestingly the scaling accuracy improved when imported into the RTP system. This improvement of geometric scaling accuracy in the RTP system may be due to use of different tools or better matrix translation in the RTP system. Still the geometric scaling inaccuracy along the z axis remained in the RTP system for thicker CT slices. These scaling errors need to be considered when doing clinical treatment planning.

A PET/CT scanner acquires separate PET and CT datasets for image registration during image acquisition. Registration of the separate PET and CT datasets are done by superimposing the PET dataset over the CT dataset. This superimposed PET and CT dataset is termed the PET/CT fused image set. This PET/CT fused image set is obtained using the Syngo 3D fusion tools. Once this single PET/CT data set is made, it can not be imported by the RTP planning software, nor can it be manipulated any further in the Syngo scanner software. The registration error analysis is done by adjusting the PET and CT contrast windows of PET/CT dataset and visualizing the registration. The ^{22}Na seeds inserted in the TGM^2 phantom were visible both in CT and PET images and are helpful in visualizing the alignment of PET and CT datasets in the PET/CT fused image set. There was no misalignment in the PET and CT datasets in the scanner.

The PET and CT datasets are imported separately into the RTP system and fused using the Syntegra auto fusion tool. A misalignment around 6 mm is measured in the PET/CT fused images and required re-alignment manually. The misalignment could be due to inaccurate reading of DICOM image file header information or an inability of the Syntegra program to align them properly. This measured misalignment in the RTP system suggests the importance of process oriented QA checks from scanner to RTP system.

The geometric scaling accuracy and registration accuracy are also analyzed by independent programs written in Mathematica 5. The geometric scaling accuracy is verified by measuring the pixel size of separate PET, separate CT and PET/CT fused image matrices. The pixel sizes were measured by finding the number of pixels for a known distance. The measured pixel sizes were close to the expected values. The accuracy of pixel sizes measured from the CT only fused image was better than from the PET only fused image when compared to expected pixel size as the CT images have a better resolution than PET images. Registration error in PET/CT fused images of the scanner is also analyzed using Mathematica. This registration error is measured by finding the difference in centroids of PET and CT only fused images. The measured registration error is 1 mm or less than one pixel. The shift in image matrixes are measured in pixels and registration error less than one pixel is insignificant. Hence the proper alignment of the PET and CT dataset in the fused image sets is verified.

The resolution of the PET scanner is measured by imaging point sources positioned at various positions in the scanner field of view as directed by NEMA recommendations. Point sources of FDG in capillary tubes were inserted in a Styrofoam block at different positions and were imaged. The PET images of point sources are fitted to a Gaussian curve to get the standard deviation of the point spread function. The standard deviation is used to compute the FWHM of

the PET activity distribution and the resolution of the scanner. NEMA 2001 standards resolution formulas are used to compute the resolution of the scanner. The measured transverse resolutions were closer to the manufacturer specified resolution than the axial resolution. The axial resolution was limited in accuracy because the point source activity in the capillary tubes extended into the Z-axis and the axial extent of the capillary tube activity was more than 1 mm.

Overall the TGM^2 Phantom was successfully used to analyze CT number value, geometric scaling and registration error in the scanner software and the RTP software. ^{22}Na seeds as PET positron sources were very convenient and fast in doing the visual registration check in the TGM^2 phantom. Also the geometric scaling check and CT number checks were simple to perform. There were a few problems with image analysis due to ^{22}Na seeds having a metallic covering that caused streak metal artifacts in the CT images. These artifacts cause problems in locating the seeds in the CT image and may have caused some error in determining centroid. Also in the PET image of the ^{22}Na seeds in the TGM^2 phantom, a curving of PET activity were observed in the four corners of the phantom. This curving artifact may also have caused error in determining centroid for PET images. This curving PET artifact was also observed in non-attenuation corrected images. The model based scatter correction algorithms used to estimate the scatter correction may have over-corrected in the corners of the TGM^2 cube as the cube does not represent a typical patient body. Also the activities of ^{22}Na seeds used for the registration check were not equal. This difference in activity did not allow a common window and level for the PET contrast window for all the seeds inserted in a side of the cube. The contrast window was adjusted for each seed to visualize the registration.

The image analysis part of the QA checks using Mathematica was time consuming as it required manually selecting individual slices from series of images and running the Mathematica

program on those slices. The programs for pixel size, registration check and resolution analysis totally take approximately 1 hour. These Mathematica programs will be more effective if we modify the program for automated selection of individual slices and estimation of pixel size, registration and resolution from a series of images. The developed program can be made more user friendly by including a graphical user interface. Also preparing the point sources for NEMA resolution measurement was time consuming. Even with the best possible equipment, making the point sources in capillary tubes requires skill and practice.

The geometric scaling check, registration check and CT number check using the TGM[2] Phantom and the NEMA resolution check can be used as an effective validation tool for the PET/CT dataset. However, problems encountered during the QA process need to be fixed to make this validation QA more effective and rapid. The streak artifact found in CT images can be removed by using a non-metallic outer covering for the seeds. The outer covering can be made of acrylic or other material close to the density of water. The curving PET artifacts could be removed by avoiding inserting ^{22}Na seeds in the corners of the cube for registration check. The resolution analysis of the scanner can be simplified by having some point source resolution inserts in the phantom. These modifications to the phantom when supplemented with an automated Mathematica software program to analyze the PET and CT images for registration, pixel size and resolution analysis will simplify the PET/CT QA process and help the technologists do a better daily QA.

REFERENCES

1) Bar-Shalom, R.; Yefremov, N.; Guralnik, L.; Gaitini, D.; Frenkel, A.; Kuten, A.; Altman, H.; Keidar, Z.; Israel, O. Clinical performance of PET/CT in evaluation of cancer: Additional value for diagnostic imaging and patient management. Journal of Nuclear Medicine, 2003; Vol. (44.): 1200-1209

2) Cohade, C.; Wahl, R.L. Application of positron emmision tomography/computed tomography image fusion in clinical positron emission tomography-Clinical use, Interpretation methods, diagnostic improvements. Seminars in Nuclear Medicine, 2003; Vol (XXXIII): 228-237

3) Townsend, D.W; Beyer, T.; Blodgett, T.M. PET/CT scanners: A Hardware approach to Image Fusion. Seminars in Nuclear Medicine, 2003; Vol (XXXIII): 193-204

4) Townsend, D.W; Beyer, T. A combined PET/CT scanner: the path to true image fusion. The British Journal of Radiology, Special issue 2002; Vol (25): S24-S30

5) Patton, J. Image Fusion in Nuclear Medicine – PET/CT. Abstract ID: 7192, 2001 AAPM Annual Meeting, Salt Lake City

6) Yap, J.T. Image Reconstruction and Image fusion (PET/CT). Abstract ID: 8391, 2002 AAPM Meeting Abstract ID: 8391, Montreal

7) Beyer, T.; Townsend, D.W; Brun, T.; Kinahan, P.E; Charron, M.; Roddy, R.; Young, J.; Byars, L.; Nutt, R. A Combined PET/CT scanner for clinical oncology. Journal of Nuclear Medicine, 2000; Vol (41):1369-1379

8) Townsend, D.W; Carney, J.P.J; Yap, J.T; Hall, N.C. PET/CT today and tomorrow. Journal of Nuclear Medicine, 2004; Vol (45): 4S-14S

9) Kalabbers, B.M; De Munck, J.C.; Slotman, B.J; Bree, R.D; Hoekstra, O.S; Boellaard, R.; Lammertsma, A.A. Matching PET and CT scans of the head and neck area: Development of method and validation. Medical Physics, 2002; Vol (29), 2230-2238

10) Kipper, M.S. Clinical application of positron emission tomography. Applied Radiology, Nov 2002; www.appliedradiology.com

11) Comtat, C. PET in Oncology, 18th Annual symposium of the Belgium hospital physicist association

12) Bushberg, J.T; Seibert, J.A; Leidholdt Jr., E.M; Boone, J.M. Essential of Physics of Imaging (2nd Edition). Williams and Wilkins, Baltimore, 1994

13) Wilting, J.T. Technical aspects of spiral CT. www.medical.philips.com

14) Levin, C.S. Data Correction Methods and Image reconstruction algorithms for positron emission tomography. 2003 AAPM meeting. Continuing education, San Diego

15) Zaidi, H.; Hasegawa, B. Determination of the attenuation map in emission tomography. Journal of Nuclear Medicine, 2003; Vol (44):291-315

16) Kinahan, P.E; Hasegawa, B.H; Beyer, T. X-Ray based attenuation correction for positron emission tomography/computed tomography scanners. Seminars in Nuclear Medicine, 2003: Vol (XXXIII): 166-179

17) Kinahan, P.E; Townsend, D.W; Beyer, T.; Sashin, D. Attenuation correction for a combined 3D PET/CT scanner. Medical Physics, 1998; Vol (25): 2046-2053

18) Humm, J.L; Rosenfeld, A; Guerra, A.D. From PET detectors to PET scanners. Eurpoean Journal of Nuclear Medicine and Molecular imaging, 2003; Vol (30): 1574-1597

19) Bujenovic, L.S; Mannting, F.; Chakrabrati, R.; Ladnier, D. Artifactual 2-Deoxy-2 [18F]Fluoro-D-Glucose Localization Surrounding Metallic objects in PET/CT scanner using CT-based attenuation correction. Molecular Imaging and Biology, 2003; Vol (5): 1-3

20) Nehmeh, A.S; Erdi, E.Y; Kalaigian, H.; Kolbert, K.S; Pan, T.; Yeung, H.; Squire, O.; Sinha, A.; Larson, M.S; Humm, L.J. Correction for Oral contrast artifacts in CT attenuation-corrected PET Images obtained by combined PET/CT. Journal of Nuclear Medicine, 2003; Vol (44):1940-1944

21) Antoch, G.; Freudenberg, S.L; Egelhof, T.; Stattaus, J.; Jentzen, W.; Debatin, F.J; Bockisch, A. FocalTracer uptake: A potential artifact in contrast-enhanced dual modality PET/CT Scans. Journal of Nuclear Medicine, 2002; Vol (43): 1339-1342

22) Cohade, C.; Osman, M.; Nakamoto, Y.; Marshal, T.L; Links, M.J; Fishman, K.E; Wahl, L.R. Initial Experience with Oral Contrast in PET/CT: Phantom and Clinical Studies. Journal of Nuclear Medicine, 2003; Vol (44): 412-416

23) Kemp, B. PET Scanner quality assurance and acceptance testing. AAPM Abstract Id: 9903, 2003 AAPM Annual Meeting, San Diego

24) Eberl, S. Evaluation of PET scanners, Acceptance Testing and QC, AAPM Meeting, 2003, San Diego

25) Pelizzari, C.A; Chen, G.T; Spelbring, D.R; Weichselbaum, R.R; Chen, C.T. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. J Comput Assist Tomogr, 1989; Vol (13): 20-26.

26) Studholme, C.; Hill, D.L; Hawkes, D.J. Automated three dimensional registration of

magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures. Medical Physics, 1997; Vol (24): 25-35

27) Mutic, S.; Dempsey, J.F; Bosch, W.R; Low, D.A; Drzymala, R.E; Chao, K.S.C; Goddu, S.M; Cutler, P.D; Purdy, J.A. Multimodality image registration quality assurance for conformal three-dimensional treatment planning. International Journal of Radiation oncology and Biological Physics, 2001; Vol (51): 255-260

28) Lavely, W.C; Scarfone, C.; Cevikalp, H.; Li, R.; Byrne, D.W; Cmelak, A.J; Dawant,B.; Price, R.R; Hallahan, D.E; Fitzpatrik, J.M. Phantom validation of coregistration of PET and CT for image-guided radiotherapy. Medical Physics, 2004; Vol (31): 1083-1092

29) Vogel, W.V; Oyen, W.J.G; Barentsz, J.O; Kaanders, J.H.A.M; Corstens, F.H.M. PET/CT: Panacea or Redundancy? Journal of Nuclear Medicine**,** 2004; Vol (45): 15S-24S

30) Forster, K.M; Mawlawi, O.; Cody, D.; Steadham, R.E; Chao, C.; Guerrero, T.; Komaki, R.; Kohlmyer, S.; Macapinlac, H. Identification of Acceptance criteria and Testing of a PET/CT scanner for radiotherapy treatment planning. abstract no. 1045. Soceity of Nuclear Medicine 50th Meeting

31) Daube-Witherspoon, M.E; Karp, J.S; Casey, M.E; Difilippo, F.P; Hines, H.; Muehllehner, G.; Simcic, V.; Stearns, C.W; Adam, L.E; Kohlmyer, S.; Sossi, V. PET Performance Measuring using the Nema NU 2-2001 Standard. Journal of Nuclear Medicine, 2002; Vol (43):1398-1409

32) Erdi, Y.E; Nehmeh, S.A; Mulmix, T.; Humm, J.L; Watson, C.C. PET Performance Measurement for an LSO-based combined PET/CT scanner using the national electrical manufacturers association NU 2-2001 Standard. Journal of Nuclear Medicine, 2004; Vol (45): 813-821

33) Powers, R. Reveal HD Manual, CPS INNOVATIONS, PRODUCT SPECIFICTION ECAT PET/CT, REV: G ECN# 02-224, Effective Date: 6/4/02

34) Verrette, A.; Sajo, E. A new dosimetry method in permanent prostate implant therapy. 2003 Summer Undergraduate Research Forum, LOUISIANA STATE UNIVERSITY, 2003

35) Ledvij, M. Curve fitting made easy. http://www.aip.org/tip/INPHFA/vol-9/iss-2/p24.html

APPENDIX A
CTPIXEL.NB

Description

This program is used to find the pixel coordinates of ^{22}Na seeds visible in the CT image. The CT gray scale image is converted to a binary image having the seeds visible. The pixel number of the ^{22}Na seed white pixel region is found and pixel coordinates are calculated.

```
Needs["Graphics`Legend`"]
Needs["Graphics`Graphics`"](*initializes programs needed for graphics function*)

SetDirectory["D:\\3Dimaging\\dicomfiles\\CTSlice"]
```

```
D:\3Dimaging\dicomfiles\CTSlice
```

```
ab = Import["CTSlice5", "DICOM"](*Import image*)
```

```
- Graphics -
```

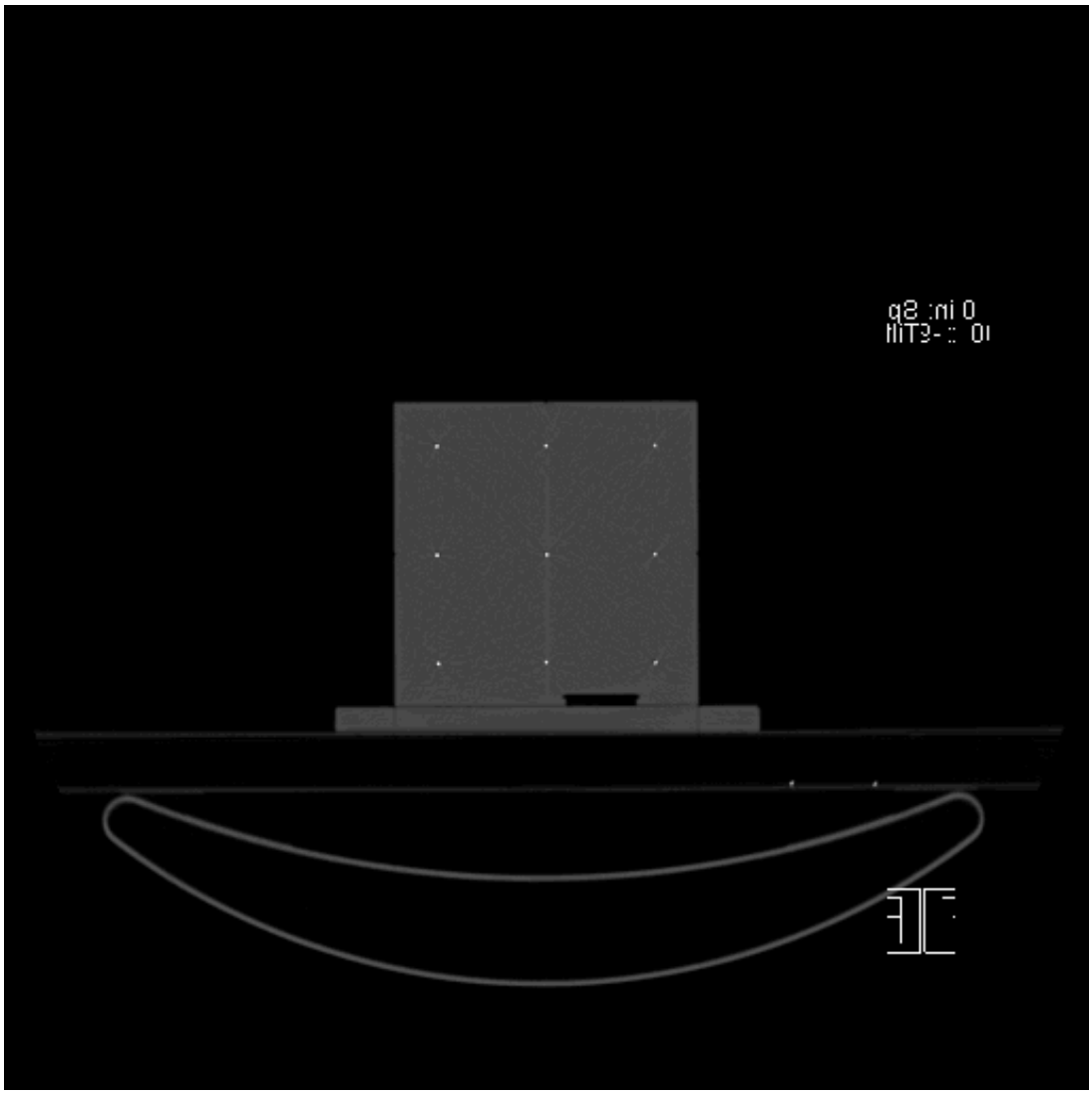
```
Dimensions[ab[[1, 4]]]
```

```
{2}
```

```
ab[[1, 4]]
```

```
ColorFunction → GrayLevel
```

```
Show[ab](* display image *)
```

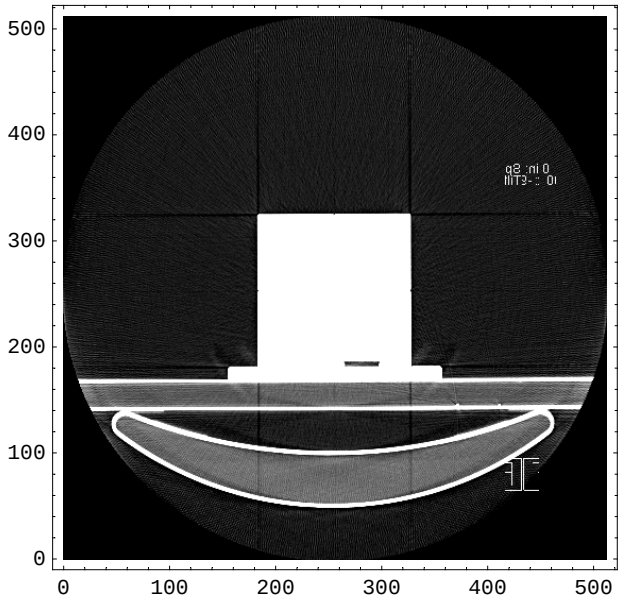


- Graphics -

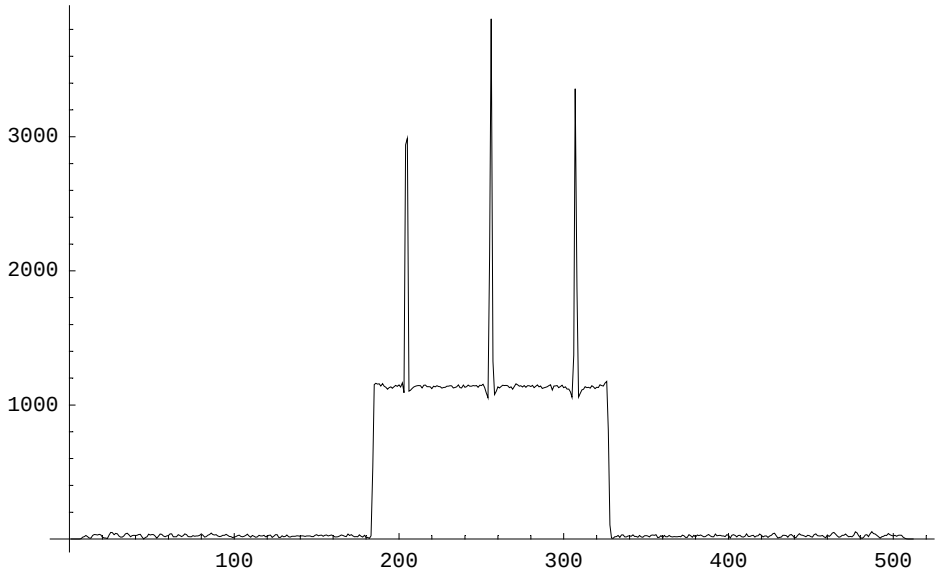
```
mygray = ab[[1, 1]];(* Store gray scale element of image in an array *)

Show[mygray];
```

Show::gtype : List is not a type of graphics. More…

```
plot2 = ListDensityPlot[mygray, Mesh → False, AspectRatio → Automatic] ;
(*Density plot of gray scale element*)
```



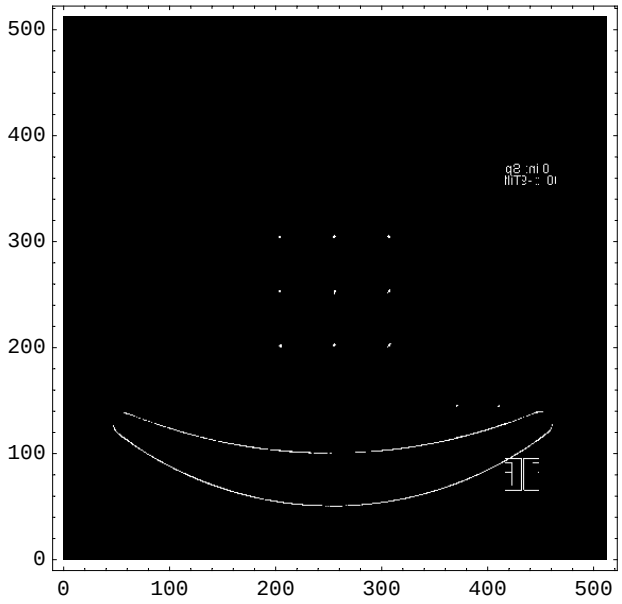

```
(* Display of grayscale values in a row containing 22 Na seeds *)
row342 = mygray[[305, All]];
xAxis = Table[i, {i, 1, Dimensions[mygray][[2]] }];
ListPlot[Transpose[{xAxis, row342}], PlotJoined → True, PlotRange → All];
```




```
(* Converting gray scale image to Binary Image *)
temp = mygray[[All, All]];
```

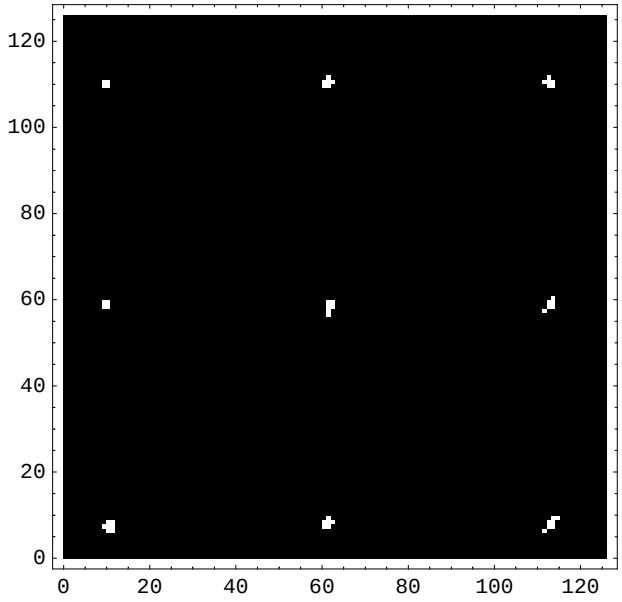
```
threshhold = 1250;
makeBinary[x_] := Module[{}, If[ (x - threshhold) ≥ 0, 1, 0] ];
binaryDICOM = Map[makeBinary, temp , {2}];
Dimensions[binaryDICOM]
plot4 = ListDensityPlot[binaryDICOM, Mesh → False, AspectRatio → Automatic];
```

```
{512, 512}
```




```
(* Cropping the Phantom region in the binary image *)

cropJPG = Table[binaryDICOM[[r, c]],
   {r, 195, 320, 1}, {c, 195, 320, 1}];
```

```
plot5 = ListDensityPlot[cropJPG, Mesh → False, AspectRatio → Automatic];
```




```
Dimensions[cropJPG]
```

```
{126, 126}
```

```
(* Summing all white pixel value in bottom row *)
```

```
acolumn1 = Table[Sum[cropJPG[[i, j]], {i, 1, 20}], {j, 1, 126}]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

```
Position[acolumn1, 0] (* Finding Pixel number of white pixel region *)
```

```
{{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {13}, {14}, {15}, {16}, {17}, {18}, {19},
 {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {44}, {45},
 {46}, {47}, {48}, {49}, {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58},
 {59}, {60}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74},
 {75}, {76}, {77}, {78}, {79}, {80}, {81}, {82}, {83}, {84}, {85}, {86}, {87},
 {88}, {89}, {90}, {91}, {92}, {93}, {94}, {95}, {96}, {97}, {98}, {99}, {100},
 {101}, {102}, {103}, {104}, {105}, {106}, {107}, {108}, {109}, {110}, {111},
 {116}, {117}, {118}, {119}, {120}, {121}, {122}, {123}, {124}, {125}, {126}}
```

```
column1pixel = {10, 11, 12, 61, 62, 63, 112, 113, 114, 115}
```

```
{10, 11, 12, 61, 62, 63, 112, 113, 114, 115}
```

```
column1pixel = column1pixel + 194 (* Actual pixel numbers of 22 Na seeds *)
```

```
{204, 205, 206, 255, 256, 257, 306, 307, 308, 309}
```

```
column1value =
 {1, 3, 3, 2, 3, 1, 1, 2, 3, 1}(* These values give weight for pixel number above *)
```

{1, 3, 3, 2, 3, 1, 1, 2, 3, 1}

```
(* Summing all white pixel value in middle row *)
acolumn2 = Table[Sum[cropJPG[[i, j]], {i, 50, 70}], {j, 1, 126}]


Position[acolumn2, 0](* Finding Pixel number of white pixel region *)
column2pixel = {10, 11, 62, 63, 112, 113, 114}
```

{10, 11, 62, 63, 112, 113, 114}

```
column2pixel = column2pixel + 194(* Actual pixel numbers of 22 Na seeds *)
```

{204, 205, 256, 257, 306, 307, 308}

```
column2value = {2, 2, 4, 2, 1, 2, 3}
```

{2, 2, 4, 2, 1, 2, 3}

```
(* Summing all white pixel value in Top row *)
acolumn3 = Table[Sum[cropJPG[[i, j]], {i, 100, 120}], {j, 1, 126}]
Position[acolumn3, 0](* Finding Pixel number of white pixel region *)
column3pixel = {10, 11, 61, 62, 63, 112, 113, 114}
```

{10, 11, 61, 62, 63, 112, 113, 114}

```
column3pixel = column3pixel + 194(* Actual pixel numbers of 22 Na seeds *)
```

{204, 205, 255, 256, 257, 306, 307, 308}

```
column3value = {2, 2, 2, 3, 1, 1, 3, 2}
```

{2, 2, 2, 3, 1, 1, 3, 2}

```
(* Summing all white pixel value in Left Column *)
arow1 = Table[Sum[cropJPG[[i, j]], {j, 1, 20}], {i, 1, 126}]
Position[arow1, 0]
row1pixel = {7, 8, 9, 59, 60, 110, 111}
```

{7, 8, 9, 59, 60, 110, 111}

```
row1pixel = row1pixel + 194
```

{201, 202, 203, 253, 254, 304, 305}

```
row1value = {2, 3, 2, 2, 2, 2, 2}
```

```
{2, 3, 2, 2, 2, 2, 2}
```

```
(* Summing all white pixel value in middle Column *)
```

```
arow2 = Table[Sum[cropJPG[[i, j]], {j, 50, 70}], {i, 1, 126}]
```

```
Position[arow2, 0]
```

```
row2pixel = {8, 9, 10, 57, 58, 59, 60, 110, 111, 112}
```

```
{8, 9, 10, 57, 58, 59, 60, 110, 111, 112}
```

```
row2pixel = row2pixel + 194
```

```
{202, 203, 204, 251, 252, 253, 254, 304, 305, 306}
```

```
row2value = {2, 3, 1, 1, 1, 2, 2, 2, 3, 1}
```

```
{2, 3, 1, 1, 1, 2, 2, 2, 3, 1}
```

```
(* Summing all white pixel value in Right Column *)
```

```
arow3 = Table[Sum[cropJPG[[i, j]], {j, 100, 120}], {i, 1, 126}]
```

```
Position[arow3, 0]
```

```
row3pixel = {7, 8, 9, 10, 58, 59, 60, 61, 110, 111, 112}
```

```
{7, 8, 9, 10, 58, 59, 60, 61, 110, 111, 112}
```

```
row3pixel = row3pixel + 194
```

```
{201, 202, 203, 204, 252, 253, 254, 255, 304, 305, 306}
```

```
row3value = {1, 2, 2, 2, 1, 2, 2, 1, 2, 3, 1}
```

APPENDIX B
FUS.NB

Description

This program is used to find the pixel coordinates of ^{22}Na seeds visible in the PET image. The 2D PET gray scale image is converted to a 1D gray scale image values. The peak value of the gray scale distribution is found to estimates the pixels coordinates. The peak values give the pixel number of ^{22}Na seeds in PET image.

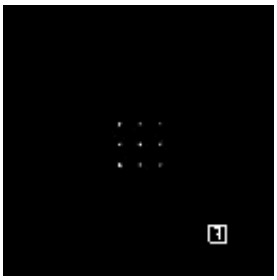
```
Needs["Graphics`Legend`"] (*loads program files for function*)
Needs["Graphics`Graphics`"]
Needs["Statistics`NonlinearFit`"]
Needs["Statistics`DiscreteDistributions`"]
Needs["Statistics`HypothesisTests`"]
Needs["Statistics`NormalDistribution`"]
Needs["Statistics`DescriptiveStatistics`"]

SetDirectory["D:\\3Dimaging\\dicomfiles\\TGMCenter\\PETSLICE128"]
(*set the directory to image file directory*)
```

```
D:\3Dimaging\dicomfiles\TGMCenter\PETSLICE128
```

```
FileNames[](*List all the files in the directory*)
```

```
{2d3d.nb, Data.doc, fus.nb, Pet1.csv, Pet2.csv, petmatrix.xls, Petpixelsize.nb,
 Petslice1, Petslice1.jpg, Petslice2, Petslice2.jpg, Petslice3, Petslice3.jpg,
 Petslice4, Petslice4.jpg, Petslice5, Petslice5.jpg, Petslice6, Petslice6.jpg}
```

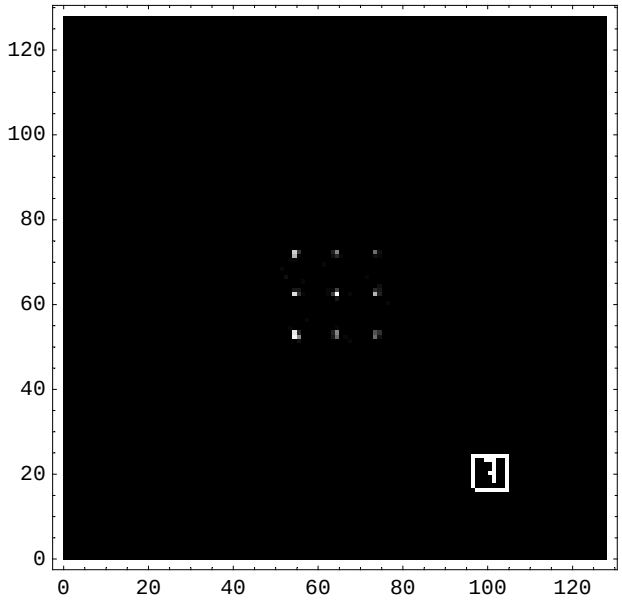
```
ab1 = Import["Petslice2", "DICOM"](*Importing medical"DICOM" image in to an array*)

ab2 = Import["Petslice3", "DICOM"]

ab3 = Import["Petslice4", "DICOM"]

ab4 = Import["Petslice5", "DICOM"]

- Graphics -

Show[ab1] (*Display the image*)
```



```
- Graphics -

mygray = ab1[[1, 1]] ;(*Taking gray scale value from image*)

Dimensions[mygray]
```

```
{128, 128}
```

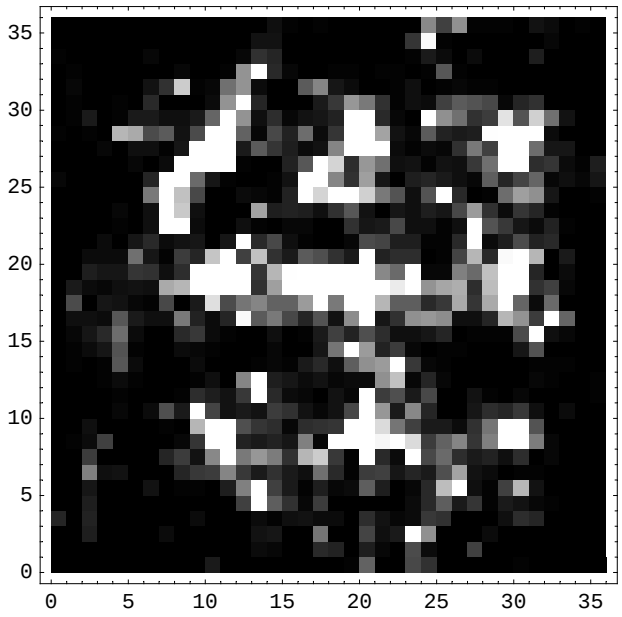
```
plot2 = ListDensityPlot[mygray, Mesh → False, AspectRatio → Automatic] ;
(*Density plot of gray scale values*)
```




```
cropJPG = Table[mygray[[r, c]],
    {r, 45, 80, 1}, {c, 45, 80, 1}] ;
 (*croping the grayscale image for the phantom object*)

Dimensions[cropJPG]
```

```
{36, 36}
```

```
plot5 = ListDensityPlot[cropJPG, Mesh → False, AspectRatio → Automatic] ;
(*Density plot of cropped region*)
```

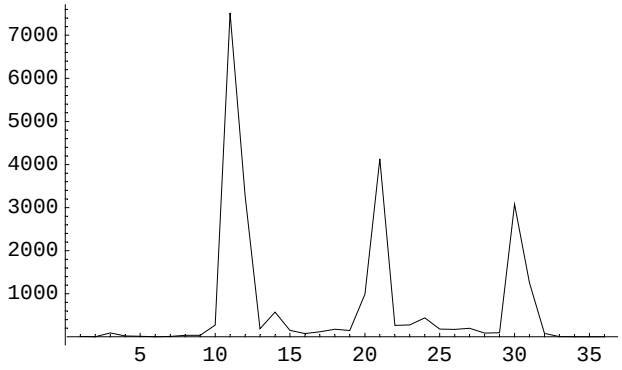


```
acolumn1 = Table[Sum[cropJPG[[i, j]], {i, 1, 13}], {j, 1, 36}]
 (*summing up bottom row of 2 D distribution to 1 D distribution*)
```

```
{10, 1, 88, 22, 11, 1, 8, 35, 37, 273, 7519, 3273, 185, 571, 153, 74, 116, 175,
 147, 989, 4130, 264, 274, 441, 179, 172, 198, 85, 95, 3077, 1255, 79, 6, 3, 0, 2}
```

```
xAxis = Table[i, {i, 1, Dimensions[acolumn1][[1]] }];
ListPlot[Transpose[{xAxis, acolumn1}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```

```
General::spell1 :
 Possible spelling error: new symbol name "xAxis" is similar to existing symbol "Axis". More…
```




```
columnpeak = { 7519, 4130, 3077}(*peak values in the 1 D distribution*)
```

```
{7519, 4130, 3077}
```

```
Table[Position[acolumn1, columnpeak[[i]]],
 {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{11}}, {{21}}, {{30}}}
```

```
acolumn1pixel = {11, 21, 30}
```

```
{11, 21, 30}
```

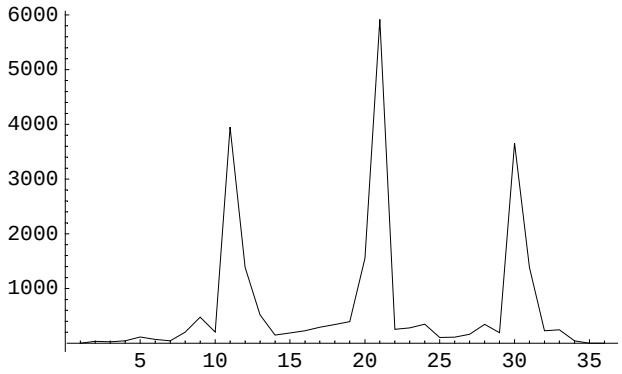
```
acolumn1pixel = acolumn1pixel + 44(*finding actual pixel number of peak values*)
```

```
{55, 65, 74}
```

```
acolumn2 = Table[Sum[cropJPG[[i, j]], {i, 14, 23}], {j, 1, 36}]
 (*summing up middle row of 2 D distribution to 1 D distribution*)
```

```
{0, 35, 27, 41, 116, 71, 43, 199, 477, 203, 3950, 1384, 520, 150, 188, 228, 290, 343, 395,
 1551, 5920, 254, 279, 348, 105, 113, 163, 347, 193, 3658, 1385, 226, 247, 43, 0, 0}
```

```
xAxis = Table[i, {i, 1, Dimensions[acolumn2][[1]] }];
ListPlot[Transpose[{xAxis, acolumn2}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```




```
columnpeak = { 3950, 5920, 3658}(*peak values in the 1 D distribution*)
```

```
{3950, 5920, 3658}
```

```
Table[Position[acolumn2, columnpeak[[i]]],
 {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{11}}, {{21}}, {{30}}}
```

```
acolumn2pixel = {11, 21, 30}
```

```
{11, 21, 30}
```

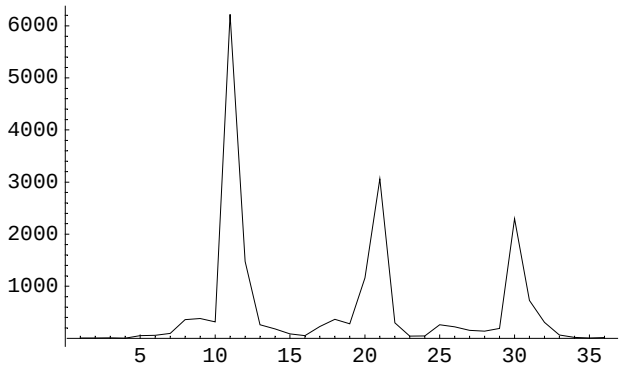
```
acolumn2pixel = acolumn2pixel + 44(*finding the actual pixel number*)
```

```
{55, 65, 74}
```

```
acolumn3 = Table[Sum[cropJPG[[i, j]], {i, 24, 36}], {j, 1, 36}]
 (*summing up Top row of 2 D distribution to 1 D distribution*)
```

```
{4, 5, 8, 2, 52, 58, 95, 360, 383, 315, 6221, 1477, 259, 180, 85, 54, 230, 366, 281,
 1160, 3063, 300, 45, 47, 262, 222, 153, 138, 190, 2298, 727, 307, 67, 18, 4, 15}
```

```
xAxis = Table[i, {i, 1, Dimensions[acolumn3][[1]] }];
ListPlot[Transpose[{xAxis, acolumn3}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```




```
columnpeak = { 6221, 3063, 2298}(*peak values in the 1 D distribution*)
```

```
{6221, 3063, 2298}
```

```
Table[Position[acolumn3, columnpeak[[i]]],
 {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{11}}, {{21}}, {{30}}}
```

```
acolumn3pixel = {11, 21, 30}
```

```
{11, 21, 30}
```

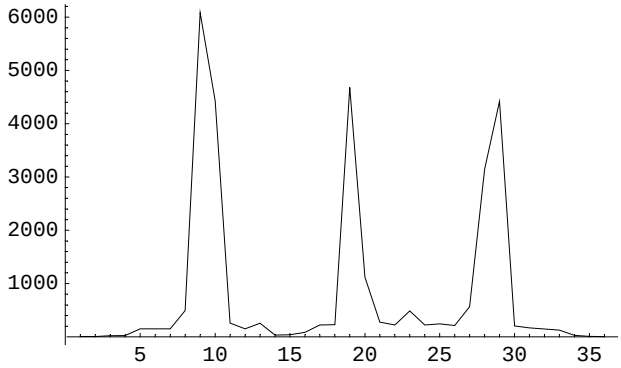
```
acolumn3pixel = acolumn3pixel + 44(*finding the actual pixel number*)
```

```
{55, 65, 74}
```

```
arow1 = Table[Sum[cropJPG[[i, j]], {j, 1, 15}], {i, 1, 36}]
 (*summing up Left column of 2 D distribution to 1 D distribution*)
```

```
{7, 5, 22, 26, 153, 153, 154, 492, 6082, 4426, 258, 153, 256, 34, 41, 85, 225, 231, 4683,
 1118, 277, 224, 486, 225, 242, 214, 564, 3146, 4419, 207, 168, 149, 128, 28, 10, 4}
```

```
xAxis = Table[i, {i, 1, Dimensions[arow1][[1]] }];
ListPlot[Transpose[{xAxis, arow1}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```




```
rowpeak = {6082, 4683, 4419}(*peak values in the 1 D distribution*)
```

```
{6082, 4683, 4419}
```

```
Table[Position[arow1, rowpeak[[i]]], {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{9}}, {{19}}, {{29}}}
```

```
arow1pixel = {9, 19, 29}
```

```
{9, 19, 29}
```

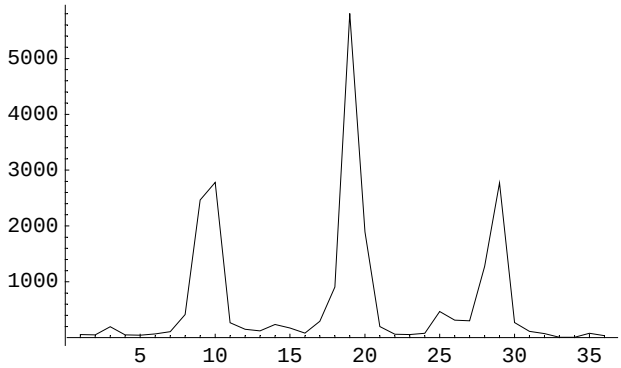
```
arow1pixel = arow1pixel + 44(*finding the actual pixel number*)
```

```
{53, 63, 73}
```

```
arow2 = Table[Sum[cropJPG[[i, j]], {j, 16, 25}], {i, 1, 36}]
 (*summing up middle column of 2 D distribution to 1 D distribution*)
```

```
{59, 53, 198, 52, 45, 67, 109, 415, 2465, 2782, 270, 151, 123, 236, 175, 81, 293, 906,
 5806, 1895, 202, 62, 57, 80, 468, 313, 303, 1277, 2773, 273, 115, 73, 7, 8, 80, 38}
```

```
xAxis = Table[i, {i, 1, Dimensions[arow2][[1]] }];
ListPlot[Transpose[{xAxis, arow2}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```



```
rowpeak = { 2782, 5806, 2773}(*peak values in the 1 D distribution*)
```

```
{2782, 5806, 2773}
```

```
Table[Position[arow2, rowpeak[[i]]], {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{10}}, {{19}}, {{29}}}
```

```
arow2pixel = {9.5, 19, 29}
```

```
{9.5, 19, 29}
```

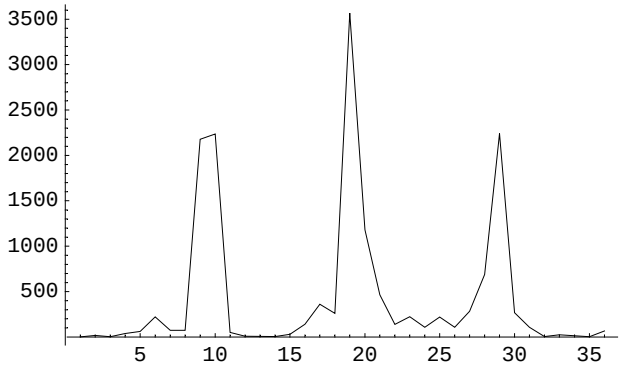
```
arow2pixel = arow2pixel + 44(*finding the actual pixel number*)
```

```
{53.5, 63, 73}
```

```
arow3 = Table[Sum[cropJPG[[i, j]], {j, 26, 36}], {i, 1, 36}]
 (*summing up middle column of 2 D distribution to 1 D distribution*)
```

```
{1, 17, 3, 39, 62, 223, 74, 74, 2178, 2236, 50, 9, 6, 4, 30, 140, 362, 261, 3568,
 1180, 466, 139, 225, 107, 221, 107, 283, 690, 2242, 269, 107, 5, 24, 14, 3, 67}
```

```
xAxis = Table[i, {i, 1, Dimensions[arow3][[1]] }];
ListPlot[Transpose[{xAxis, arow3}], PlotJoined → True, PlotRange → All] ;
(*Plot of 1 D activity and pixel number*)
```




```
rowpeak = { 2236, 3568, 2242}(*peak values in the 1 D distribution*)
```

```
{2236, 3568, 2242}
```

```
Table[Position[arow3, rowpeak[[i]]], {i, 1, 3}](*finding pixels having peak values*)
```

```
{{{10}}, {{19}}, {{29}}}
```

```
arow3pixel = {9.5, 19, 29}
```

```
{9.5, 19, 29}
```

```
arow3pixel = arow3pixel + 44(*finding the actual pixel number*)
```

```
{53.5, 63, 73}
```

APPENDIX C
PETCTFUS.NB

Description

This program is used to find the pixel coordinates of ^{22}Na seeds visible in TGM^2 CT only fused image and PET only fused image. The CT only gray scale image is converted to a binary image having the seeds visible. The pixel number of the ^{22}Na seed white pixel region is found and pixel coordinates are calculated. In the 2D PET only image, the 2D gray scale image is converted to a 1D gray scale image values. The peak value of the gray scale distribution is found by nonlinear curve fitting of Gaussian distribution on the gray scale distribution. The estimated center of the distribution gives the coordinates of ^{22}Na seeds.

```
(* Initializes program needed for non linear fit and Graphics *)
Needs["Graphics`Legend`"]
Needs["Graphics`Graphics`"]
Needs["Statistics`NonlinearFit`"]
Needs["Statistics`DiscreteDistributions`"]
Needs["Statistics`HypothesisTests`"]
Needs["Statistics`NormalDistribution`"]
Needs["Statistics`DescriptiveStatistics`"]

SetDirectory["D:\\3Dimaging\\dicomfiles\\PETCT"]
```

```
D:\3Dimaging\dicomfiles\PETCT
```

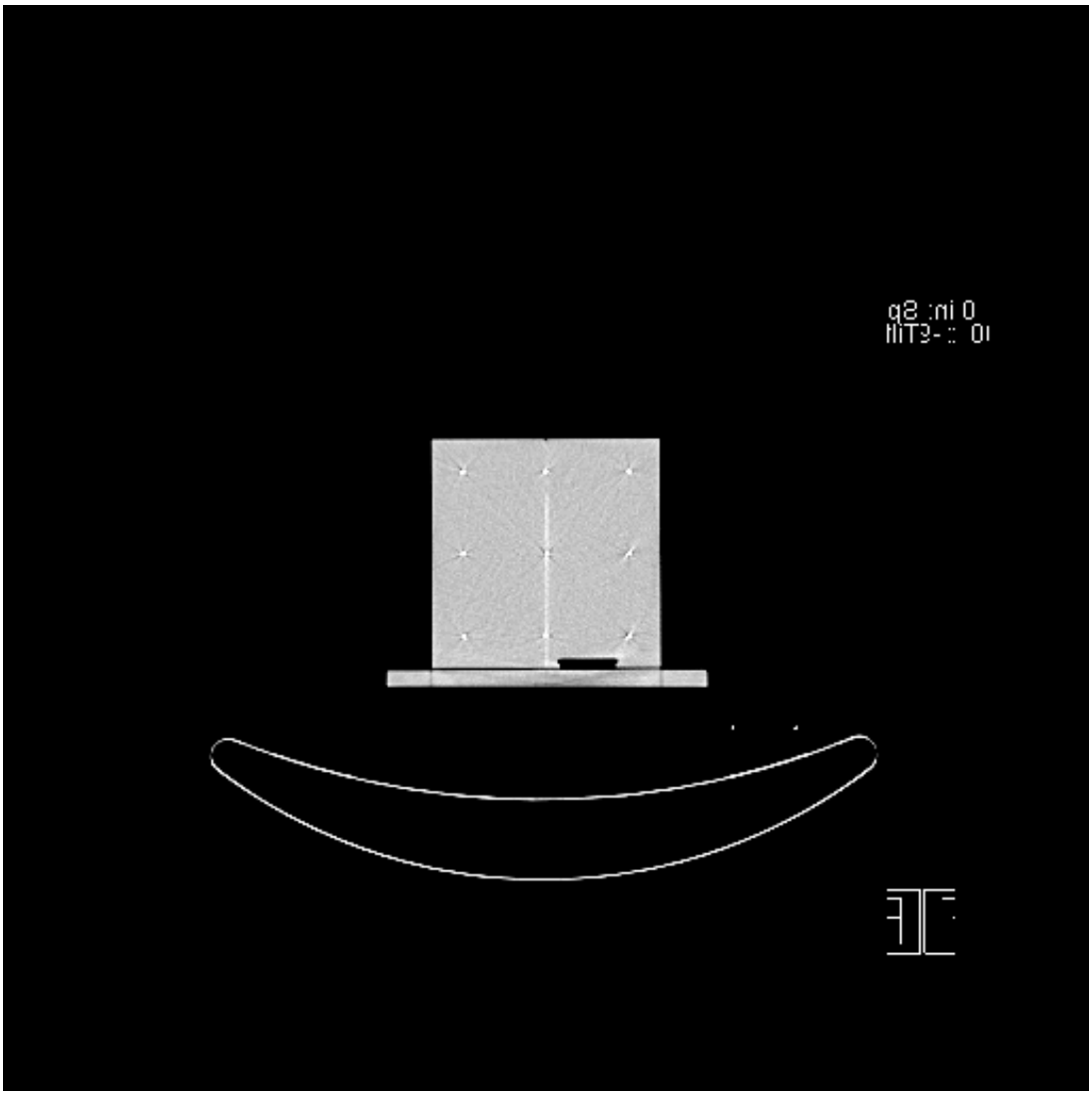
```
FileNames[]
```

```
{91584563, 91584563.jpg, 91586634, 91586634.jpg, 91586657, 91586657.jpg, 91586680,
 91586680.jpg, 91586703, 91586703.jpg, 91586726, 91586726.jpg, 91586749, 91586749.jpg,
 91586772, 91586772.jpg, 91586795, 91586795.jpg, 91586818, 91586818.jpg,
 91586841, 91586841.jpg, 91586864, 91586864.jpg, 91586887, 91586887.jpg,
 91586910, 91586910.jpg, data.doc, pet_ct_fusioncheck.xls, petctfusion.nb}
```

```
(* Import CT only fused PET/CT Image *)
ab = Import["91586795", "DICOM"]
```

```
- Graphics -
```

```
Show[ab]
```

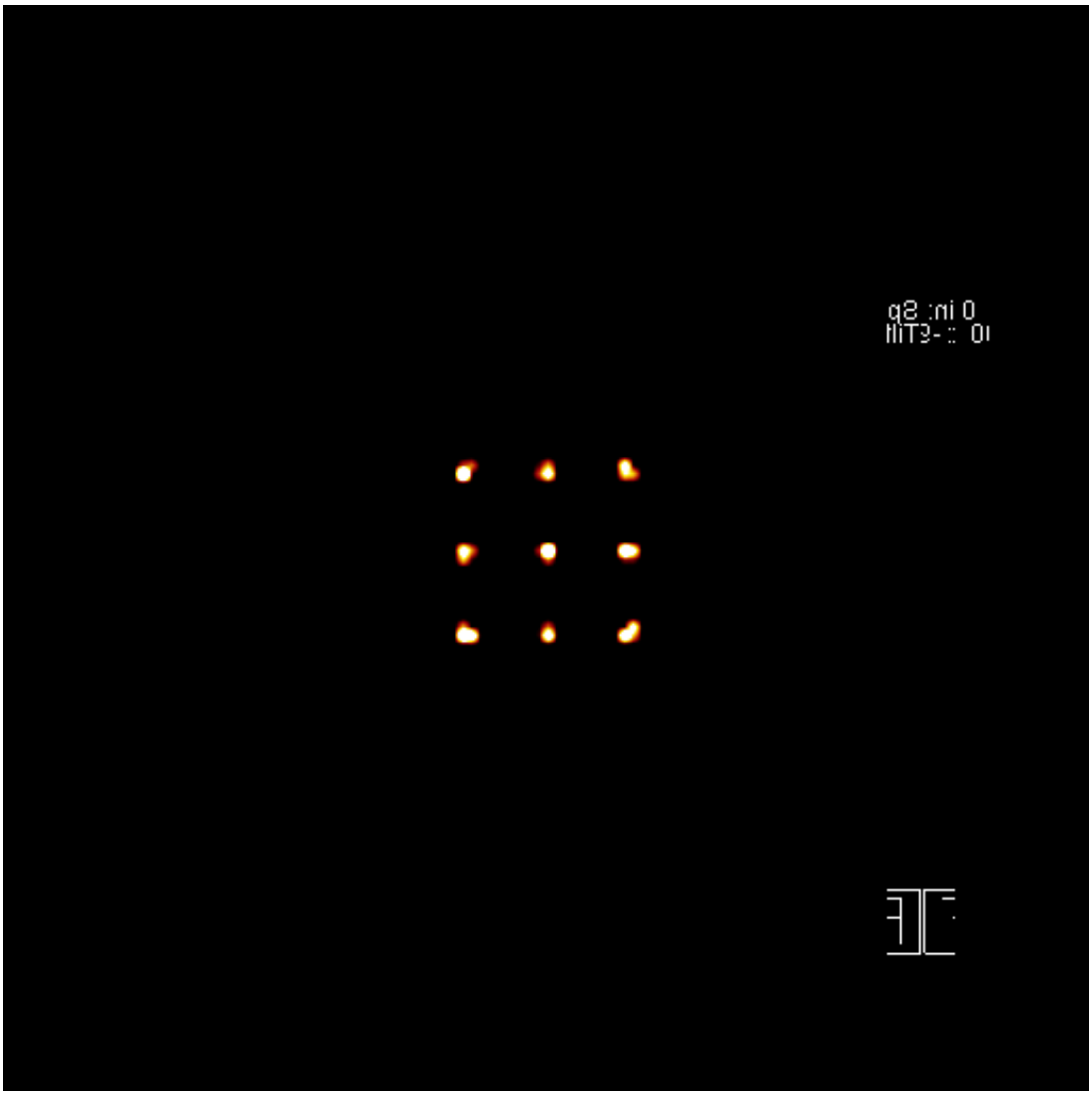


```
- Graphics -
```

```
(*Import PET only PET/CT Fused image *)
ab1 = Import["91586818", "DICOM"]
```

```
- Graphics -
```

```
Show[ab1]
```



```
- Graphics -
```

```
temp = ab[[1, 1]];
mygray = temp[[All, All, 3]];
```

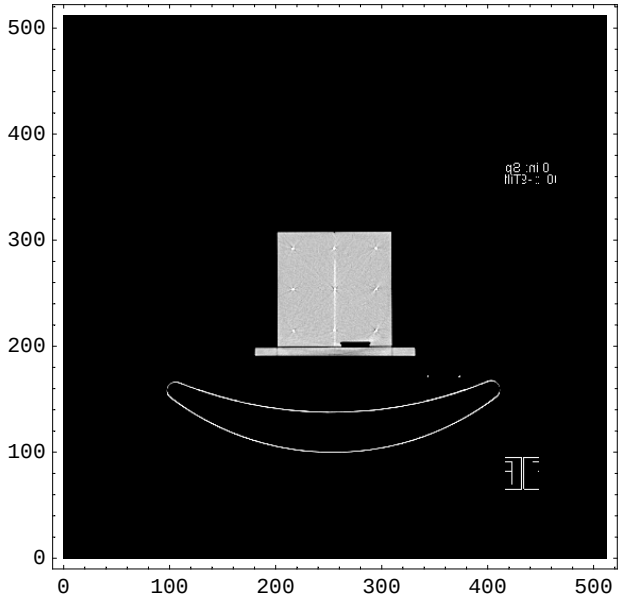
```
mygray = Sum[temp[[All, All, i]], {i, 1, 3}];
```

```
Dimensions[mygray]
```

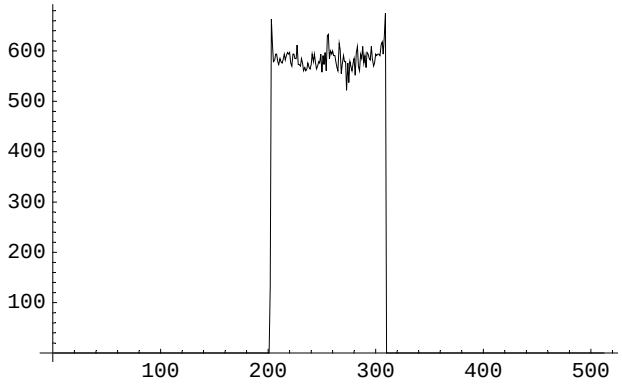
```
{512, 512}
```

```
(* Density plot of gray scale values of CT Image *)
```

```
plot2 = ListDensityPlot[mygray, Mesh → False, AspectRatio → Automatic];
```




```
row342 = mygray[[305, All]];
xAxis = Table[i, {i, 1, Dimensions[mygray][[2]]}];
ListPlot[Transpose[{xAxis, row342}], PlotJoined → True, PlotRange → All];
```

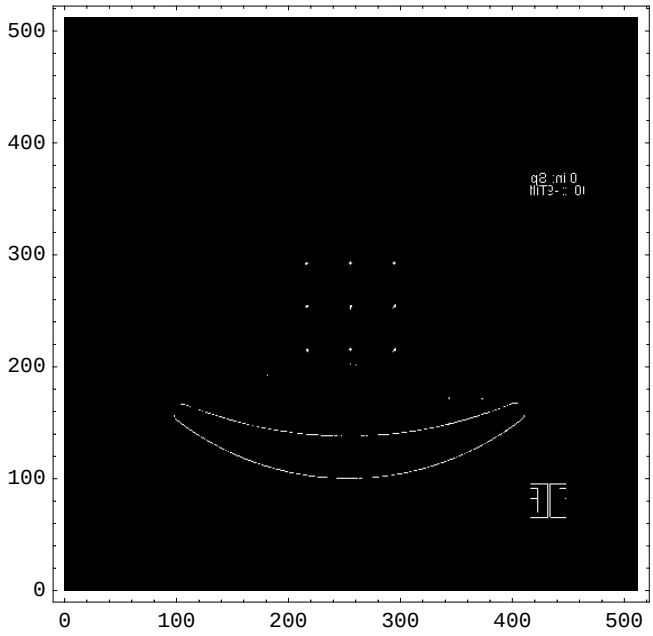



```
(* Conversion of gray scale image to binary image *)
temp = mygray[[All, All]];

Dimensions[temp]
```

```
{512, 512}
```

```
threshhold = 750;
makeBinary[x_] := Module[{}, If[ (x - threshhold) ≥ 0, 1, 0] ];
binaryDICOM = Map[makeBinary, temp , {2}];
Dimensions[binaryDICOM]
plot4 = ListDensityPlot[binaryDICOM, Mesh → False, AspectRatio → Automatic];
```

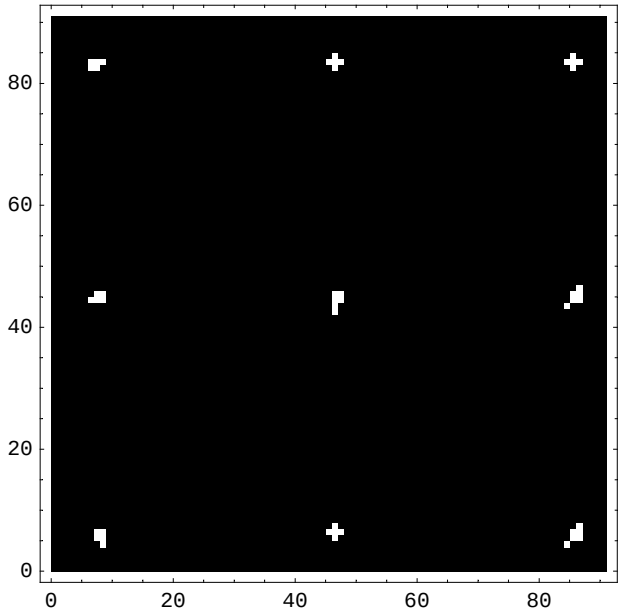
```
{512, 512}
```




```
(* Cropping of Phantom region in binary image *)

cropJPG = Table[binaryDICOM[[r, c]],
   {r, 210, 300, 1}, {c, 210, 300, 1}];

Dimensions[cropJPG]
```

```
{91, 91}
```

```
plot5 = ListDensityPlot[cropJPG, Mesh → False, AspectRatio → Automatic];
```




```
(* Summation of binary values in bottom row *)
acolumn1 = Table[Sum[cropJPG[[i, j]], {i, 1, 15}], {j, 1, 91}]
```

```
{0, 0, 0, 0, 0, 0, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0}
```

```
Position[acolumn1, 0](* Finding white pixel column number along the row *)
```

```
{{1}, {2}, {3}, {4}, {5}, {6}, {7}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17},
 {18}, {19}, {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31},
 {32}, {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {44}, {45},
 {49}, {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75},
 {76}, {77}, {78}, {79}, {80}, {81}, {82}, {83}, {84}, {88}, {89}, {90}, {91}}
```

```
column1pixel = {8, 9, 46, 47, 48, 85, 86, 87}
```

```
{8, 9, 46, 47, 48, 85, 86, 87}
```

```
column1pixel = column1pixel + 209(* Finding actual pixel number *)
```

```
{217, 218, 255, 256, 257, 294, 295, 296}
```

```
column1value = { 2, 3, 1, 3, 1, 1, 2, 3}
```

```
{2, 3, 1, 3, 1, 1, 2, 3}
```

```
(* Summation of binary values in middle row *)
```

```
acolumn2 = Table[Sum[cropJPG[[i, j]], {i, 40, 50}], {j, 1, 91}]
```

```
{0, 0, 0, 0, 0, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0}
```

```
Position[acolumn2, 0](* Finding white pixel column number along the row *)
```

```
{{1}, {2}, {3}, {4}, {5}, {6}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18},
 {19}, {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {44}, {45}, {46},
 {49}, {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75},
 {76}, {77}, {78}, {79}, {80}, {81}, {82}, {83}, {84}, {88}, {89}, {90}, {91}}
```

```
column2pixel = {7, 8, 9, 47, 48, 85, 86, 87}
```

```
{7, 8, 9, 47, 48, 85, 86, 87}
```

```
column2pixel = column2pixel + 209(* Finding actual pixel number *)
```

```
{216, 217, 218, 256, 257, 294, 295, 296}
```

```
column2value = { 1, 2, 2, 4, 2, 1, 2, 3}
```

```
{1, 2, 2, 4, 2, 1, 2, 3}
```

```
{1, 2, 2, 4, 2, 1, 2, 3}
```

```
(* Summation of binary values in top row *)
```

```
acolumn3 = Table[Sum[cropJPG[[i, j]], {i, 75, 90}], {j, 1, 91}]
```

```
{0, 0, 0, 0, 0, 0, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0}
```

```
Position[acolumn3, 0](* Finding white pixel column number along the row *)
```

```
{{1}, {2}, {3}, {4}, {5}, {6}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18},
 {19}, {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {44}, {45}, {49},
 {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75},
 {76}, {77}, {78}, {79}, {80}, {81}, {82}, {83}, {84}, {88}, {89}, {90}, {91}}
```

```
column3pixel = {7, 8, 9, 46, 47, 48, 85, 86, 87}
```

```
{7, 8, 9, 46, 47, 48, 85, 86, 87}
```

```
column3pixel = column3pixel + 209 (* Finding actual pixel number *)
```

```
{216, 217, 218, 255, 256, 257, 294, 295, 296}
```

```
column3value = {2, 2, 1, 1, 3, 1, 1, 3, 1}
```

```
{2, 2, 1, 1, 3, 1, 1, 3, 1}
```

```
(* Summation of binary values in left column *)
arow1 = Table[Sum[cropJPG[[i, j]], {j, 1, 15}], {i, 1, 91}]
```

```
{0, 0, 0, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0}
```

```
Position[arow1, 0](* Finding white pixel row number along the column *)
```

```
{{1}, {2}, {3}, {4}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18},
 {19}, {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {44}, {47}, {48},
 {49}, {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75}, {76},
 {77}, {78}, {79}, {80}, {81}, {82}, {85}, {86}, {87}, {88}, {89}, {90}, {91}}
```

```
row1pixel = {5, 6, 7, 45, 46, 83, 84}
```

```
{5, 6, 7, 45, 46, 83, 84}
```

```
row1pixel = row1pixel + 209 (* Finding actual pixel number *)
```

```
{214, 215, 216, 254, 255, 292, 293}
```

```
row1value = {1, 2, 2, 3, 2, 2, 3}
```

```
{1, 2, 2, 3, 2, 2, 3}
```

```
(* Summation of binary values in middle column *)
arow2 = Table[Sum[cropJPG[[i, j]], {j, 40, 55}], {i, 1, 91}]
```

```
{0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0, 0, 0}
```

```
Position[arow2, 0] (* Finding white pixel row number along the column *)
```

```
{{1}, {2}, {3}, {4}, {5}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18},
 {19}, {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {47}, {48}, {49},
 {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75},
 {76}, {77}, {78}, {79}, {80}, {81}, {82}, {86}, {87}, {88}, {89}, {90}, {91}}
```

```
row2pixel = {6, 7, 8, 43, 44, 45, 46, 83, 84, 85}
```

```
{6, 7, 8, 43, 44, 45, 46, 83, 84, 85}
```

```
row2pixel = row2pixel + 209 (* Finding actual pixel number *)
```

```
{215, 216, 217, 252, 253, 254, 255, 292, 293, 294}
```

```
row2value = {1, 3, 1, 1, 1, 2, 2, 1, 3, 1}
```

```
{1, 3, 1, 1, 1, 2, 2, 1, 3, 1}
```

```
(* Summation of binary values in right column *)
arow3 = Table[Sum[cropJPG[[i, j]], {j, 70, 90}], {i, 1, 91}]
```

```
{0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 0, 0, 0, 0, 0}
```

```
Position[arow3, 0](* Finding white pixel row number along the column *)
```

```
{{1}, {2}, {3}, {4}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {19},
 {20}, {21}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {31}, {32},
 {33}, {34}, {35}, {36}, {37}, {38}, {39}, {40}, {41}, {42}, {43}, {48}, {49},
 {50}, {51}, {52}, {53}, {54}, {55}, {56}, {57}, {58}, {59}, {60}, {61}, {62},
 {63}, {64}, {65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75},
 {76}, {77}, {78}, {79}, {80}, {81}, {82}, {86}, {87}, {88}, {89}, {90}, {91}}
```

```
row3pixel = {5, 6, 7, 8, 44, 45, 46, 47, 83, 84, 85}
```

```
{5, 6, 7, 8, 44, 45, 46, 47, 83, 84, 85}
```

```
row3pixel = row3pixel + 209 (* Finding actual pixel number *)
```

```
{214, 215, 216, 217, 253, 254, 255, 256, 292, 293, 294}
```

```
row3value = {1, 2, 2, 1, 1, 2, 2, 1, 1, 3, 1}
```

```
{1, 2, 2, 1, 1, 2, 2, 1, 1, 3, 1}
```

```
(* PET Image Centroid *)
temp = ab1[[1, 1]];
```

```
Dimensions[temp]
```

```
{512, 512, 3}
```

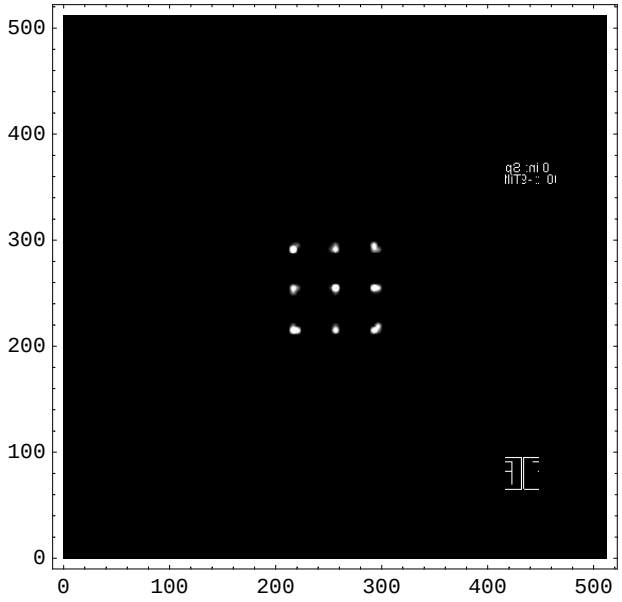
```
mygray1 = temp[[All, All, 1]];
```

```
mygray1 = Sum[temp[[All, All, i]], {i, 1, 3}];
```

```
Show[mygray1];
```

```
(* Density plot of gray scale image of PET *)
```

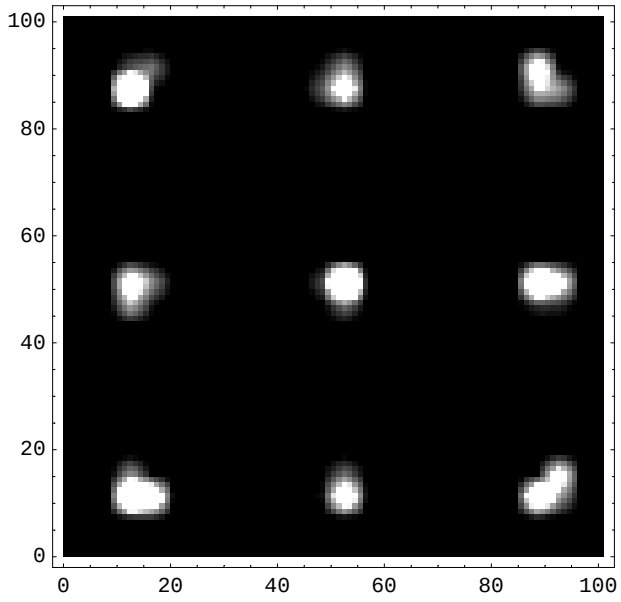
```
plot6 = ListDensityPlot[mygray1, Mesh → False, AspectRatio → Automatic];
```




```
(* cropping of Phantom region from PET Image *)
cropJPG = Table[mygray1[[r, c]],
   {r, 205, 305, 1}, {c, 205, 305, 1}];

Dimensions[cropJPG]
```

```
{101, 101}
```

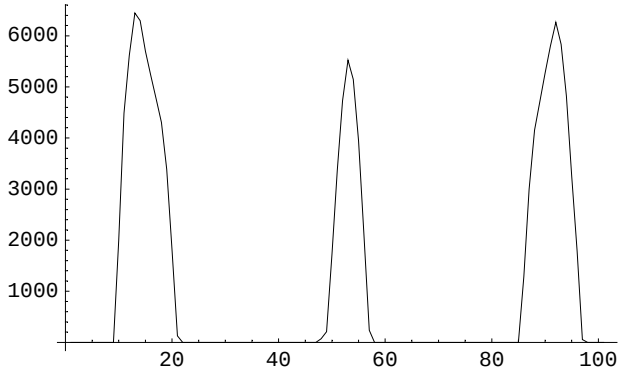
```
plot5 = ListDensityPlot[cropJPG, Mesh → False, AspectRatio → Automatic];
```



```
(* Summation of bottom row of 2 D PET activity to 1 D along a row *)
acolumn1 = Table[Sum[cropJPG[[i, j]], {i, 1, 30}], {j, 1, 101}]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1993, 4487, 5613, 6446, 6294, 5697, 5229, 4766,
 4304, 3386, 1817, 126, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 69, 206, 1730, 3365, 4730, 5520, 5146, 3944, 2141, 236, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1285,
 2999, 4155, 4711, 5269, 5798, 6266, 5833, 4820, 3262, 1784, 53, 0, 0, 0, 0}
```

```
(* Plot of PET activity along the row *)

xAxis = Table[i, {i, 1, Dimensions[acolumn1][[1]] }];
ListPlot[Transpose[{xAxis, acolumn1}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian Fitting on the above plot *)
data = Transpose[ {xAxis, acolumn1} ];

TableForm[ data ];

Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 6000}, {position1, 15}, {sigma1, 2},
   {amp2, 5000}, {position2, 55}, {sigma2, 2},
   {amp3, 6000}, {position3, 90}, {sigma3, 2},
   {Offset, 10}},
  ShowProgress -> False, MaxIterations → 500]

(* Pixel numbers of the Gaussian peak *)
column1pixel = {14.5469, 53.1433, 91.2948}
```

```
{14.5469, 53.1433, 91.2948}
```

```
column1pixel = column1pixel + 204 (* Actual pixel numbers *)
```

```
{218.547, 257.143, 295.295}
```

```
stdcolumn1pixel = {14.3344, 14.7595, 52.9512, 53.3354, 91.0814, 91.5081}
```

```
{14.3344, 14.7595, 52.9512, 53.3354, 91.0814, 91.5081}
```

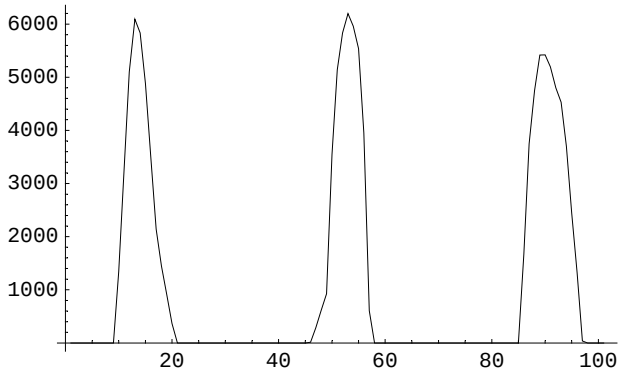
```
stdcolumn1pixel = stdcolumn1pixel + 204
```

```
{218.334, 218.76, 256.951, 257.335, 295.081, 295.508}
```

```
(* Summation of middle row of 2 D PET activity to 1 D along a row *)
acolumn2 = Table[Sum[cropJPG[[i, j]], {i, 40, 60}], {j, 1, 101}]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1342, 3229, 5103, 6094, 5835, 4882, 3518, 2141, 1450,
 913, 365, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 15, 287, 609, 926, 3544, 5144, 5835, 6198, 5958, 5540, 3951, 607, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1667,
 3747, 4748, 5419, 5421, 5196, 4801, 4530, 3699, 2448, 1319, 38, 0, 0, 0, 0}
```

```
(* Plot of PET activity along the row *)
xAxis = Table[i, {i, 1, Dimensions[acolumn2][[1]] }];
ListPlot[Transpose[{xAxis, acolumn2}], PlotJoined → True, PlotRange → All];
```



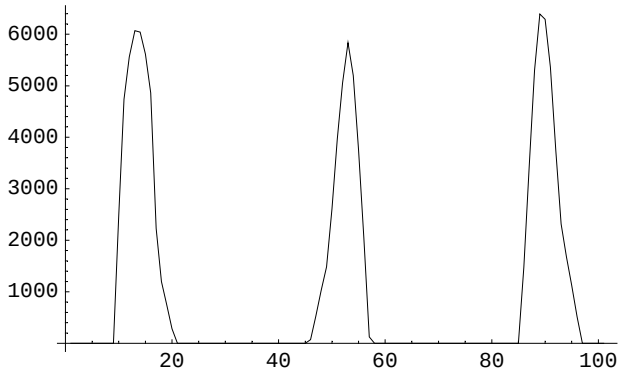

```
(* Nonlinear Gaussian Fitting on the above plot *)
```

```
data = Transpose[ {xAxis, acolumn2} ];
```

```
TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 6000}, {position1, 15}, {sigma1, 2},
   {amp2, 6000}, {position2, 55}, {sigma2, 2},
   {amp3, 5000}, {position3, 90}, {sigma3, 2},
   {Offset, 10}},
  ShowProgress -> False, MaxIterations → 500]


(* Pixel numbers of the Gaussian peak *)

column2pixel = {13.6705, 53.0352, 90.5817}

{13.6705, 53.0352, 90.5817}

column2pixel = column2pixel + 204 (* Actual pixel numbers *)

{217.671, 257.035, 294.582}

stdcolumn2pixel = {13.494, 13.847, 52.8716, 53.1987, 90.3666, 90.7969}

{13.494, 13.847, 52.8716, 53.1987, 90.3666, 90.7969}

stdcolumn2pixel = stdcolumn2pixel + 204

{217.494, 217.847, 256.872, 257.199, 294.367, 294.797}

(* Summation of top row of 2 D PET activity to 1 D along a row *)
acolumn3 = Table[Sum[cropJPG[[i, j]], {i, 80, 100}], {j, 1, 101}]

{0, 0, 0, 0, 0, 0, 0, 0, 0, 2468, 4737, 5560, 6071, 6042, 5612, 4861, 2228,
 1199, 749, 284, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 71, 530, 1023, 1483, 2613, 3963, 5054, 5834, 5194, 3751, 2047, 124,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1474, 3400, 5302, 6394, 6289, 5360, 3734, 2318, 1685, 1126, 511, 0, 0, 0, 0, 0}


(* Plot of PET activity along the row *)
```

```
xAxis = Table[i, {i, 1, Dimensions[acolumn3][[1]] }];
ListPlot[Transpose[{xAxis, acolumn3}], PlotJoined → True, PlotRange → All];
```

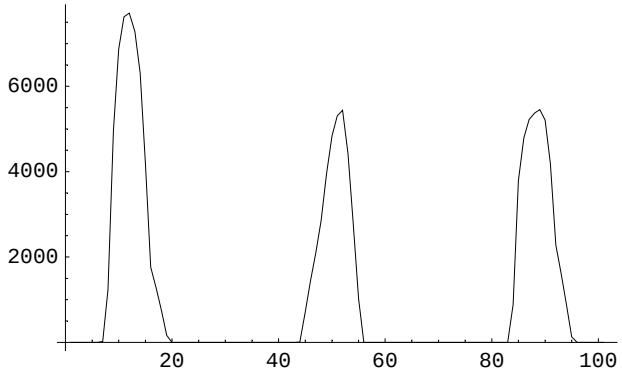



```
(* Nonlinear Gaussian Fitting on the above plot *)
data = Transpose[ {xAxis, acolumn3} ];

TableForm[ data ];

Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 6000}, {position1, 15}, {sigma1, 2},
   {amp2, 6000}, {position2, 55}, {sigma2, 2},
   {amp3, 6000}, {position3, 90}, {sigma3, 2},
   {Offset, 10}},
  ShowProgress -> False, MaxIterations → 500]


(* Pixel numbers of the Gaussian peak *)
column3pixel = {13.5936, 52.7933, 89.7217}

{13.5936, 52.7933, 89.7217}

column3pixel = column3pixel + 204(* Actual pixel numbers *)

{217.594, 256.793, 293.722}

stdcolumn3pixel = {13.4537, 13.7335, 52.6417, 52.9449, 89.5843, 89.8592}

{13.4537, 13.7335, 52.6417, 52.9449, 89.5843, 89.8592}

stdcolumn3pixel = stdcolumn3pixel + 204

{217.454, 217.734, 256.642, 256.945, 293.584, 293.859}
```

```
(* Summation of left column of 2 D PET activity to 1 D along a column *)
arow1 = Table[Sum[cropJPG[[i, j]], {j, 1, 30}], {i, 1, 101}]
```

```
{0, 0, 0, 0, 0, 0, 22, 1238, 4963, 6876, 7626, 7717, 7282, 6312, 4179, 1761,
 1273, 746, 163, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 6, 715, 1449, 2103, 2851, 3954, 4832, 5309, 5437, 4423, 2763, 1030, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 883,
 3789, 4790, 5222, 5369, 5451, 5205, 4186, 2280, 1617, 889, 130, 0, 0, 0, 0, 0, 0}
```

```
(* Plot of PET activity along the column *)
xAxis = Table[i, {i, 1, Dimensions[arow1][[1]] }];
ListPlot[Transpose[{xAxis, arow1}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian Fitting on the above plot *)
data = Transpose[ {xAxis, arow1} ];

TableForm[ data ];

Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 7000}, {position1, 15}, {sigma1, 2},
   {amp2, 5000}, {position2, 50}, {sigma2, 2},
   {amp3, 5000}, {position3, 90}, {sigma3, 2},
   {Offset, 10}},
  ShowProgress -> False, MaxIterations → 500]


(* Pixel numbers of the Gaussian peak *)
row1pixel = {11.9648, 50.874, 88.3815}
```

```
{11.9648, 50.874, 88.3815}
```

```
row1pixel = row1pixel + 204(* Actual pixel numbers *)
```

```
{215.965, 254.874, 292.382}
```

```
stdrow1pixel = {11.828, 12.1015, 50.6618, 51.0863, 88.1787, 88.5842}
```

```
{11.828, 12.1015, 50.6618, 51.0863, 88.1787, 88.5842}
```

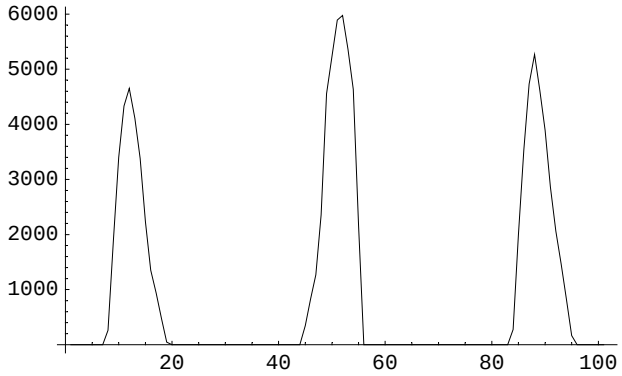
```
stdrow1pixel = stdrow1pixel + 204
```

```
{215.828, 216.102, 254.662, 255.086, 292.179, 292.584}
```

```
(* Summation of middle column of 2 D PET activity to 1 D along a column *)
arow2 = Table[Sum[cropJPG[[i, j]], {j, 40, 60}], {i, 1, 101}]
```

```
{0, 0, 0, 0, 0, 0, 0, 262, 1905, 3399, 4332, 4650, 4108, 3381, 2231, 1355, 938,
 479, 47, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 347, 829, 1270, 2354, 4555, 5235, 5890, 5976, 5361, 4633, 2164, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 278, 1989,
 3537, 4730, 5265, 4613, 3890, 2862, 2062, 1468, 829, 164, 0, 0, 0, 0, 0, 0}
```

```
(* Plot of PET activity along the column *)
xAxis = Table[i, {i, 1, Dimensions[arow2][[1]] }];
ListPlot[Transpose[{xAxis, arow2}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian Fitting on the above plot *)
data = Transpose[ {xAxis, arow2} ];

TableForm[ data ];

Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 5000}, {position1, 15}, {sigma1, 2},
   {amp2, 7000}, {position2, 50}, {sigma2, 2},
   {amp3, 6000}, {position3, 90}, {sigma3, 2},
   {Offset, 10}},
  ShowProgress -> False, MaxIterations → 500]
```

```
(* Pixel numbers of the Gaussian peak *)
row2pixel = {12.1821, 51.3937, 88.4301}
```

```
{12.1821, 51.3937, 88.4301}
```

```
row2pixel = row2pixel + 204 (* Actual pixel numbers *)
```

```
{216.182, 255.394, 292.43}
```

```
stdrow2pixel = {12.003, 12.3612, 51.2575, 51.5299, 88.2605, 88.5997}
```

```
{12.003, 12.3612, 51.2575, 51.5299, 88.2605, 88.5997}
```

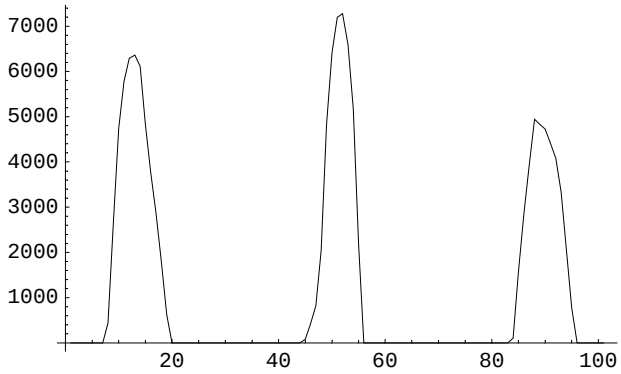
```
stdrow2pixel = stdrow2pixel + 204
```

```
{216.003, 216.361, 255.258, 255.53, 292.261, 292.6}
```

```
(* Summation of Right column of 2 D PET activity to 1 D along a column *)
arow3 = Table[Sum[cropJPG[[i, j]], {j, 80, 100}], {i, 1, 101}]
```

```
{0, 0, 0, 0, 0, 0, 0, 438, 2686, 4742, 5778, 6290, 6360, 6113, 4817, 3770, 2839,
 1778, 624, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 80, 424, 818, 2027, 4831, 6410, 7197, 7280, 6596, 5175, 2195, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 104, 1560,
 2840, 3909, 4943, 4832, 4728, 4417, 4081, 3337, 2059, 783, 0, 0, 0, 0, 0, 0}
```

```
(* Plot of PET activity along the column *)
xAxis = Table[i, {i, 1, Dimensions[arow3][[1]] }];
ListPlot[Transpose[{xAxis, arow3}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian Fitting on the above plot *)
data = Transpose[ {xAxis, arow3} ];

TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ] +
   amp2 × Exp[ -(x - position2)^2 / (2 × sigma2^2) ] + amp3 × Exp[ -(x - position3)^2 / (2 × sigma3^2) ] + Offset,
  {x},
  {{amp1, 6000}, {position1, 15}, {sigma1, 2},
   {amp2, 7000}, {position2, 50}, {sigma2, 2},
   {amp3, 5000}, {position3, 90}, {sigma3, 2},
   {Offset, 0}},
  ShowProgress -> False, MaxIterations → 500]

(* Pixel numbers of the Gaussian peak *)
row3pixel = {12.9264, 51.5168, 89.6327}
```

```
{12.9264, 51.5168, 89.6327}
```

```
row3pixel = row3pixel + 204 (* Actual pixel numbers *)
```

```
{216.926, 255.517, 293.633}
```

```
stdrow3pixel = {12.7624, 13.0905, 51.3917, 51.6419, 89.4177, 89.8477}
```

```
{12.7624, 13.0905, 51.3917, 51.6419, 89.4177, 89.8477}
```

```
stdrow3pixel = stdrow3pixel + 204
```

```
{216.762, 217.091, 255.392, 255.642, 293.418, 293.848}
```

APPENDIX D
TRANSRES.NB

Description

This program is used to find the FWHM of point sources in PET image. The 2D gray scale PET image of the point sources is converted to 1D gray scale distribution. Then the 1D gray scale distribution is curve fitted using a Gaussian curve distribution. The estimated best Gaussian curve fit gives the FWHM of the point source distribution.

```
Needs["Graphics`Legend`"]
Needs["Graphics`Graphics`"]
Needs["Statistics`NonlinearFit`"]
Needs["Statistics`DiscreteDistributions`"]
Needs["Statistics`HypothesisTests`"]
Needs["Statistics`NormalDistribution`"]
Needs["Statistics`DescriptiveStatistics`"]

SetDirectory["D://3DImaging//FBP//nema512//transverse//4cm"]
Directory[]
```

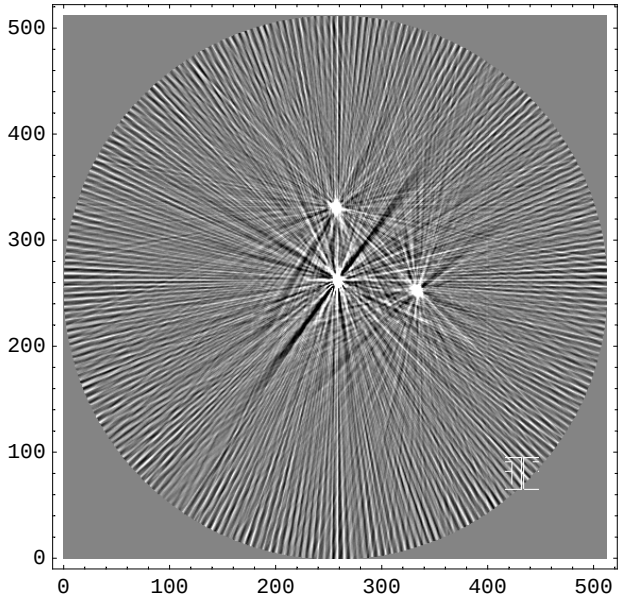
```
D:\3DImaging\FBP\nema512\transverse\4cm
```

```
D:\3DImaging\FBP\nema512\transverse\4cm
```

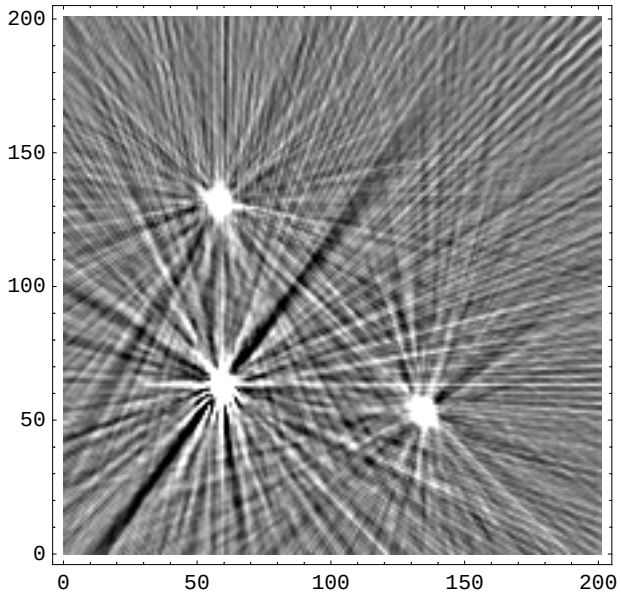
```
FileNames[]
```

```
{1335294, 1335294.jpg, 1335319, 1335319.jpg, 1335344, 1335344.jpg, 1390667, 1390667.jpg,
 1390692, 1390692.jpg, 1390717, 1390717.jpg, 1390742, 1390742.jpg, 1390767, 1390767.jpg}
```

```
(* Importing all seven PET Transverse Image slices *)

ab1 = Import["1335294", "DICOM"];
ab2 = Import["1335319", "DICOM"];
ab3 = Import["1335344", "DICOM"];
ab4 = Import["1390692", "DICOM"];
ab5 = Import["1390717", "DICOM"];
ab6 = Import["1390742", "DICOM"];
ab7 = Import["1390767", "DICOM"];

(* Summing up all seven slices/Reducing axial dimension *)
ab = ab1[[1, 1]] + ab2[[1, 1]] +
   ab3[[1, 1]] + ab4[[1, 1]] + ab5[[1, 1]] + ab6[[1, 1]] + ab7[[1, 1]];

mygray = ab;

plot2 = ListDensityPlot[mygray, Mesh → False, AspectRatio → Automatic];
```




```
(* Cropping the Point source region *)
cropJPG = Table[ab[[r, c]],
   {r, 200, 400, 1}, {c, 200, 400, 1}];

plot5 = ListDensityPlot[cropJPG, Mesh → False, AspectRatio → Automatic];
```




```
Dimensions[cropJPG]
```

```
{201, 201}
```

```
(* Finding Resy at (10,0) center of FOV *)
aresyx10y0 = Table[Sum[cropJPG[[i, j]], {j, 120, 155}], {i, 20, 100}]
```

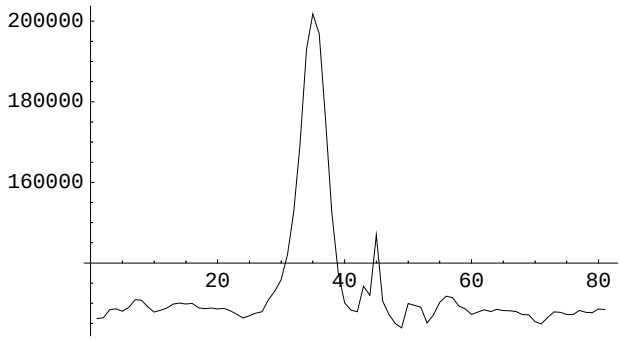
```
{126209, 126408, 128447, 128597, 128002, 128986, 130851, 130756, 129104, 127804, 128282,
 128802, 129809, 130044, 129782, 129942, 128850, 128624, 128819, 128568, 128716,
 128158, 127260, 126377, 126897, 127516, 127879, 130864, 133009, 135871, 142123,
 152743, 169396, 193089, 201781, 196937, 175969, 152459, 137540, 130107, 128311,
 127893, 134221, 131987, 146949, 130490, 127233, 124957, 123897, 129915, 129522,
 129006, 125118, 127101, 130223, 131726, 131383, 129337, 128600, 127216, 127801,
 128392, 127917, 128525, 128171, 128132, 127941, 127194, 127138, 125429, 124899,
 126496, 127840, 127724, 127196, 127188, 128224, 127747, 127659, 128543, 128488}
```

```
Dimensions[aresyx10y0]
```

```
Min[aresyx10y0]
```

```
123897
```

```
(* Plot of activity along y direction *)
xAxis = Table[i, {i, 1, Dimensions[aresyx10y0][[1]] }];
ListPlot[Transpose[{xAxis, aresyx10y0}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian fit for the above plot *)
```

```
data = Transpose[ {xAxis, aresyx10y0} ];
```

```
TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 200000}, {position1, 35}, {sigma1, 2},
   {Offset, 120000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 73818.7, position1 → 35.0525, sigma1 → 2.01454, Offset → 128671.},
```

ParameterCITable →

	Estimate	Asymptotic SE	CI
amp1	73818.7	1852.96	{70129., 77508.5}
position1	35.0525	0.0579026	{34.9372, 35.1678},
sigma1	2.01454	0.0593551	{1.89634, 2.13273}
Offset	128671.	338.15	{127998., 129344.}

EstimatedVariance → 8.03709×10^{6},

ANOVATable →

	DF	SumOfSq	MeanSq
Model	4	1.45644×10^{12}	3.64109×10^{11}
Error	77	6.18856×10^{8}	8.03709×10^{6} ,
Uncorrected Total	81	1.45706×10^{12}	
Corrected Total	80	1.83607×10^{10}	

AsymptoticCorrelationMatrix →

$$\begin{pmatrix} 1. & -1.51317 \times 10^{-13} & -0.53014 & -0.129041 \\ -1.51317 \times 10^{-13} & 1. & 8.48113 \times 10^{-14} & 3.99156 \times 10^{-15} \\ -0.53014 & 8.48113 \times 10^{-14} & 1. & -0.219874 \\ -0.129041 & 3.99156 \times 10^{-15} & -0.219874 & 1. \end{pmatrix},$$

FitCurvatureTable →

	Curvature
Max Intrinsic	0.102612
Max Parameter-Effects	0.135432
95. % Confidence Region	0.633667

}

```
(* Resy (10,0) = 2.01452 *)

(* Finding Resx at (10,0) center of FOV *)
aresxx10y0 = Table[Sum[cropJPG[[i, j]], {i, 20, 100}], {j, 120, 155}]
```

```
General::spell1 :
 Possible spelling error: new symbol name "aresxx10y0" is similar to existing symbol "aresyx10y0". More…
```

```
{286705, 287494, 287501, 287219, 288633, 290842, 288981, 287050, 289074,
 292378, 296251, 306781, 318139, 336007, 350900, 362719, 359033, 341234,
 321344, 302120, 298579, 295366, 289641, 288135, 289871, 287983, 286109,
 286204, 287220, 288315, 287358, 285695, 285278, 285433, 286304, 287210}
```

```
Dimensions[aresxx10y0]
```

```
{36}
```

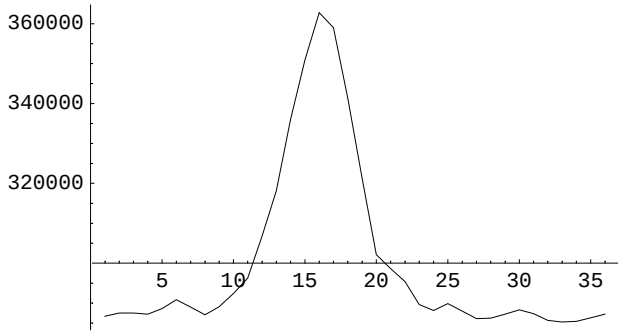
```
Max[aresxx10y0]
```

362719

```
Min[aresxx10y0]
```

285278

```
(* Plot of activity along x direction *)
xAxis = Table[i, {i, 1, Dimensions[aresxx10y0][[1]] }];
ListPlot[Transpose[{xAxis, aresxx10y0}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear Gaussian curve fitting for above plot *)

data = Transpose[ {xAxis, aresxx10y0} ];

TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 360000}, {position1, 15}, {sigma1, 2},
   {Offset, 300000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 73577.1, position1 → 16.108, sigma1 → 2.36555, Offset → 287745.},
```

ParameterCITable →

	Estimate	Asymptotic SE	CI
amp1	73577.1	1203.46	{71125.7, 76028.4}
position1	16.108	0.0434014	{16.0196, 16.1964},
sigma1	2.36555	0.0471263	{2.26956, 2.46154}
Offset	287745.	403.873	{286923., 288568.}

EstimatedVariance → 3.82036×10^{6},

ANOVATable →

	DF	SumOfSq	MeanSq
Model	4	3.25448×10^{12}	8.13619×10^{11}
Error	32	1.22252×10^{8}	3.82036×10^{6} ,
Uncorrected Total	36	3.2546×10^{12}	
Corrected Total	35	1.75333×10^{10}	

AsymptoticCorrelationMatrix →

$$\begin{pmatrix} 1. & 1.42897 \times 10^{-12} & -0.424061 & -0.237301 \\ 1.42897 \times 10^{-12} & 1. & 4.48043 \times 10^{-13} & -2.36778 \times 10^{-12} \\ -0.424061 & 4.48043 \times 10^{-13} & 1. & -0.38966 \\ -0.237301 & -2.36778 \times 10^{-12} & -0.38966 & 1. \end{pmatrix},$$

FitCurvatureTable →

	Curvature
Max Intrinsic	0.0697227
Max Parameter-Effects	0.0911022 }
95. % Confidence Region	0.612169

```
(* Resx at (10,0) = 2.36555 *)


(* finding Resy at (0,1) center of fov *)

aresyx0y1 = Table[Sum[cropJPG[[i, j]], {j, 20, 80}], {i, 20, 100}]
```

```
{214938, 215088, 214947, 214416, 215145, 216434, 217325, 216834, 218045, 218628, 217374,
 216681, 216576, 215443, 215361, 216717, 217157, 216907, 215452, 214146, 214986,
 216929, 217869, 217757, 217863, 218207, 217453, 216982, 217777, 219087, 217311,
 218438, 215617, 214615, 217000, 223543, 225302, 225527, 225323, 226307, 239473,
 262455, 300209, 384626, 480161, 412429, 318125, 265705, 229978, 225315, 223724,
 219826, 215257, 213453, 213910, 214391, 215529, 216884, 217895, 218135, 216954,
 216520, 217335, 217347, 215928, 214350, 213836, 213852, 214244, 215374, 216819,
 218411, 219856, 220653, 220722, 219165, 218052, 217762, 217704, 218219, 218432}
```

```
Dimensions[aresyx0y1]
```

```
{81}
```

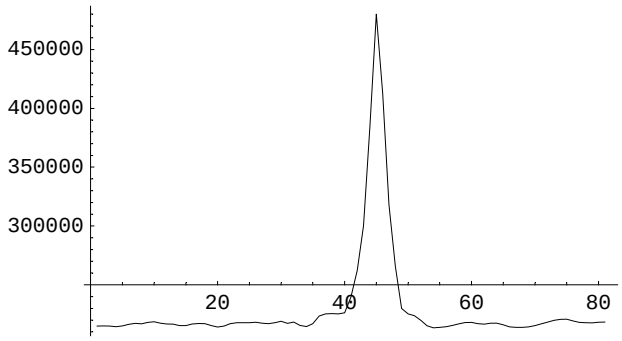
```
Max[aresyx0y1]
```

```
480161
```

```
Min[aresyx0y1]
```

```
213453
```

```
(* Plot of activity along y direction *)
xAxis = Table[i, {i, 1, Dimensions[aresyx0y1][[1]] }];
ListPlot[Transpose[{xAxis, aresyx0y1}], PlotJoined → True, PlotRange → All];
```




```
(* Non linear gaussian fit for above plot *)
data = Transpose[ {xAxis, aresyx0y1} ];
```

```
TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 460000}, {position1, 45}, {sigma1, 2},
   {Offset, 220000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 242256., position1 → 45.116, sigma1 → 1.48494, Offset → 217960.},
```

ParameterCITable →		Estimate	Asymptotic SE	CI
	amp1	242256.	4044.04	{234204., 250309.}
	position1	45.116	0.0284531	{45.0593, 45.1726},
	sigma1	1.48494	0.0289607	{1.42728, 1.54261}
	Offset	217960.	622.804	{216720., 219201.}

EstimatedVariance → 2.8356×10^{7},

ANOVATable →		DF	SumOfSq	MeanSq
	Model	4	4.3956×10^{12}	1.0989×10^{12}
	Error	77	2.18341×10^{9}	2.8356×10^{7} ,
	Uncorrected Total	81	4.39778×10^{12}	
	Corrected Total	80	1.46612×10^{11}	

AsymptoticCorrelationMatrix →

$$\begin{pmatrix} 1. & 5.30861 \times 10^{-8} & -0.543555 & -0.108898 \\ 5.30861 \times 10^{-8} & 1. & -9.26851 \times 10^{-8} & 1.64716 \times 10^{-8} \\ -0.543555 & -9.26851 \times 10^{-8} & 1. & -0.186419 \\ -0.108898 & 1.64716 \times 10^{-8} & -0.186419 & 1. \end{pmatrix},$$

FitCurvatureTable →		Curvature
	Max Intrinsic	0.0678325
	Max Parameter-Effects	0.0897581 }
	95. % Confidence Region	0.633667

```
(* Resy at (0,1) = 1.48494 *)

(* Finding Resx at (0,1) Center of FOV *)
aresxx0y1 = Table[Sum[cropJPG[[i, j]], {i, 20, 100}], {j, 20, 80}]
```

```
{287278, 288013, 288049, 288479, 288093, 287872, 287439, 287058, 285929, 283791, 283289,
 283785, 284733, 284682, 284272, 283486, 282502, 282537, 284309, 285204, 285034,
 285985, 287980, 288471, 288138, 290799, 290267, 287156, 288428, 289927, 291465,
 290681, 288639, 288650, 291414, 298914, 295871, 336432, 402831, 492632, 559189,
 465587, 391545, 343002, 308548, 294824, 292410, 294197, 295555, 291199, 286475,
 285744, 288434, 287647, 285494, 286286, 286711, 285464, 285777, 286198, 285722}
```

```
Dimensions[aresxx0y1]
```

```
{61}
```

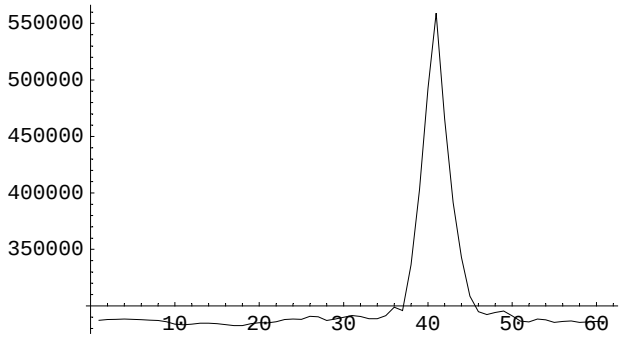
```
Max[aresxx0y1]
```

559189

```
Min[aresxx0y1]
```

282502

```
(* Plot of activity along x direction *)
xAxis = Table[i, {i, 1, Dimensions[aresxx0y1][[1]] }];
ListPlot[Transpose[{xAxis, aresxx0y1}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear gaussian curve fitting for the above plot *)
data = Transpose[ {xAxis, aresxx0y1} ];

TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 560000}, {position1, 40}, {sigma1, 2},
   {Offset, 290000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 249969., position1 → 40.9227, sigma1 → 1.58476, Offset → 287927.},

                                  Estimate     Asymptotic SE     CI
                      amp1        249969.      4303.06           {241352., 258585.}
 ParameterCITable →   position1   40.9227      0.0312243         {40.8601, 40.9852},
                      sigma1      1.58476      0.0320476         {1.52059, 1.64894}
                      Offset      287927.      804.982           {286315., 289539.}

 EstimatedVariance → 3.40672×10^7,

                                  DF      SumOfSq           MeanSq
             Model                4       5.80434×10^12     1.45109×10^12
 ANOVATable → Error               57      1.94183×10^9      3.40672×10^7 ,
             Uncorrected Total    61      5.80628×10^12
             Corrected Total      60      1.61291×10^11

 AsymptoticCorrelationMatrix →

 ⎛      1.          -2.15287×10^-9    -0.527785        -0.13228       ⎞
 ⎜ -2.15287×10^-9         1.          3.89242×10^-9   -8.44477×10^-10 ⎟
 ⎜   -0.527785     3.89242×10^-9          1.           -0.225209      ⎟
 ⎝   -0.13228     -8.44477×10^-10    -0.225209            1.         ⎠ ,

                                                 Curvature
 FitCurvatureTable → Max Intrinsic               0.070444
                     Max Parameter-Effects       0.092935 }
                     95. % Confidence Region     0.62825
```

```
(* Resx at (0.1) = 1.58476 *)


(* finding Resx at (0,10) center of fov *)

aresxx0y10 = Table[Sum[cropJPG[[i, j]], {i, 100, 200}], {j, 20, 80}]
```

```
{358929, 359289, 358375, 357775, 357930, 357871, 357883, 358495, 358790, 358674, 358423,
 358672, 358759, 359695, 360197, 360161, 360770, 362399, 362650, 359916, 358980,
 358939, 357235, 356033, 356637, 359599, 358652, 355128, 354118, 352707, 355424,
 358547, 358216, 360007, 367983, 375505, 376430, 427069, 459128, 459805, 465349,
 388731, 382733, 382523, 366440, 364881, 367398, 365001, 361003, 359639, 359744,
 358992, 359864, 359984, 358779, 359177, 359909, 358566, 357729, 358255, 358625}
```

```
Dimensions[aresxx0y10]
```

```
{61}
```

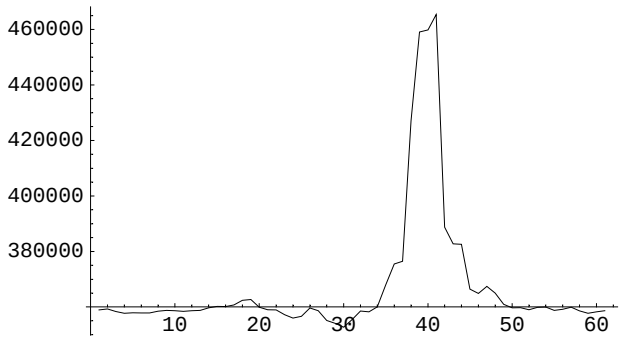
```
Max[aresxx0y10]
```

```
465349
```

```
Min[aresxx0y10]
```

```
352707
```

```
(* plot of activity along x direction *)
xAxis = Table[i, {i, 1, Dimensions[aresxx0y10][[1]] }];
ListPlot[Transpose[{xAxis, aresxx0y10}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear gaussian fitting on the above plot *)
data = Transpose[ {xAxis, aresxx0y10} ];

TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 460000}, {position1, 40}, {sigma1, 2},
   {Offset, 290000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 110255., position1 → 39.8342, sigma1 → 1.71144, Offset → 359543.},
```

ParameterCITable →

	Estimate	Asymptotic SE	CI
amp1	110255.	3920.49	{102404., 118105.}
position1	39.8342	0.0695959	{39.6948, 39.9735},
sigma1	1.71144	0.0716008	{1.56806, 1.85481}
Offset	359543.	766.461	{358008., 361078.}

EstimatedVariance → 3.04891×10^{7},

ANOVATable →

	DF	SumOfSq	MeanSq
Model	4	8.26254×10^{12}	2.06563×10^{12}
Error	57	1.73788×10^{9}	3.04891×10^{7} ,
Uncorrected Total	61	8.26428×10^{12}	
Corrected Total	60	3.49452×10^{10}	

AsymptoticCorrelationMatrix →

$$\begin{pmatrix} 1. & -8.73839 \times 10^{-11} & -0.52331 & -0.13824 \\ -8.73839 \times 10^{-11} & 1. & 1.52349 \times 10^{-10} & -3.47208 \times 10^{-11} \\ -0.52331 & 1.52349 \times 10^{-10} & 1. & -0.234991 \\ -0.13824 & -3.47208 \times 10^{-11} & -0.234991 & 1. \end{pmatrix},$$

FitCurvatureTable →

	Curvature
Max Intrinsic	0.145797
Max Parameter-Effects	0.192195
95. % Confidence Region	0.62825

}

```
(* Resx (0,10) = 1.71144 *)


(* Finding Resy (0,10) at center of FOV *)
```

```
aresyx0y10 = Table[Sum[cropJPG[[i, j]], {j, 20, 80}], {i, 100, 200}]
```

```
General::spell1 :
 Possible spelling error: new symbol name "aresyx0y10" is similar to existing symbol "aresxx0y10". More…
```

```
{218432, 217867, 217417, 217210, 218233, 219021, 218299, 218171, 217838, 217774, 217437,
 217772, 217732, 218141, 217892, 217585, 218757, 219624, 219099, 217658, 216918,
 216425, 216773, 216379, 217315, 220075, 220859, 224453, 231833, 242728, 266620,
 292688, 308875, 306499, 274142, 249649, 231363, 222818, 219534, 221079, 219699,
 217933, 217179, 216847, 216829, 216779, 217325, 217696, 217188, 216590, 216709,
 216719, 216011, 216129, 216748, 217589, 216795, 216545, 217040, 216989, 216408,
 216166, 216139, 216291, 216688, 217030, 216694, 216492, 216339, 216794, 217370,
 216843, 216307, 216205, 216596, 217262, 217371, 217048, 216972, 216687, 216960,
 217249, 217413, 217014, 216684, 216685, 216790, 216799, 216806, 216500, 216416,
 216358, 216308, 216538, 216584, 216460, 216434, 216575, 216462, 216962, 217095}
```

```
Dimensions[aresyx0y10]
```

```
{101}
```

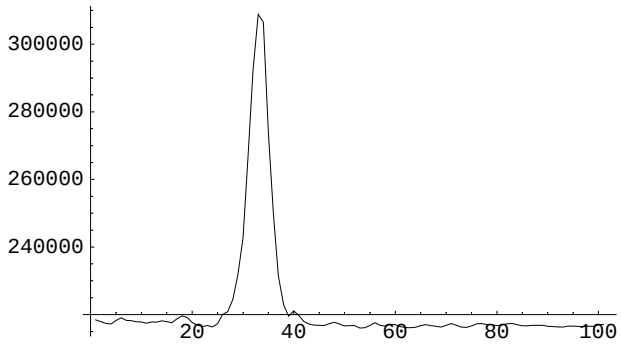
```
Max[aresyx0y10]
```

```
308875
```

```
Min[aresyx0y10]
```

```
216011
```

```
(* Plot of activity distribution along y direction *)
xAxis = Table[i, {i, 1, Dimensions[aresyx0y10][[1]] }];
ListPlot[Transpose[{xAxis, aresyx0y10}], PlotJoined → True, PlotRange → All];
```




```
(* Nonlinear gaussian fitting on the above plot *)
data = Transpose[ {xAxis, aresyx0y10} ];
```

```
TableForm[ data ];
```

```
Clear[x]
answer = NonlinearRegress[data,
  amp1 × Exp[ -(x - position1)^2 / (2 × sigma1^2) ]
  + Offset,
  {x},
  {{amp1, 300000}, {position1, 32}, {sigma1, 2},
   {Offset, 200000}},
  ShowProgress -> False, MaxIterations → 5000]
```

```
{BestFitParameters →
  {amp1 → 91971.4, position1 → 33.1687, sigma1 → 1.98842, Offset → 217294.},
```

ParameterCITable →

	Estimate	Asymptotic SE	CI
amp1	91971.4	844.415	{90295.4, 93647.3}
position1	33.1687	0.0209449	{33.1271, 33.2103}
sigma1	1.98842	0.0213491	{1.94605, 2.03079}
Offset	217294.	135.238	{217026., 217563.}

EstimatedVariance → 1.65386×10^{6},

ANOVATable →

	DF	SumOfSq	MeanSq
Model	4	4.99792×10^{12}	1.24948×10^{12}
Error	97	1.60424×10^{8}	1.65386×10^{6}
Uncorrected Total	101	4.99808×10^{12}	
Corrected Total	100	2.78917×10^{10}	

AsymptoticCorrelationMatrix →

$$\begin{pmatrix} 1. & -2.788 \times 10^{-12} & -0.54084 & -0.113248 \\ -2.788 \times 10^{-12} & 1. & 1.87676 \times 10^{-12} & -1.11254 \times 10^{-12} \\ -0.54084 & 1.87676 \times 10^{-12} & 1. & -0.193682 \\ -0.113248 & -1.11254 \times 10^{-12} & -0.193682 & 1. \end{pmatrix},$$

FitCurvatureTable →

	Curvature
Max Intrinsic	0.0373528
Max Parameter-Effects	0.0494007
95. % Confidence Region	0.636868

```
(* Resy (0,10) = 1.98842 *)
```

VITA

Rajesh Manoharan was born in Madras, India, on September 10, 1977. He attended College of Engineering, Anna University, Guindy, Tamil Nadu and received Bachelor of Engineering degree in mechanical engineering in September 1999. He came to United States in August 2000 and started his graduate studies in Industrial Engineering at the University of Toledo, Ohio. During his research work in graduate studies he got interested in radiological science. He received a master's degree in industrial engineering in May 2002. He joined Louisiana State University in August 2002 to pursue graduate studies in medical physics and health physics. He expects to receive this degree in December 2004.